



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI141502

SISTEM JEMURAN PAKAIAN OTOMATIS MENGUNAKAN MOTOR DAN SENSOR BERBASIS ARDUINO

ADETIYA BAGUS NUSANTARA
NRP 05111440000004

Dosen Pembimbing I
Ir. Muchammad Husni, M.Kom

Dosen Pembimbing II
Henning Titi Ciptaningtyas, S.Kom., M.Kom

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - KI141502

SISTEM JEMURAN PAKAIAN OTOMATIS MENGUNAKAN MOTOR DAN SENSOR BERBASIS ARDUINO

ADETIYA BAGUS NUSANTARA
NRP 05111440000044

Dosen Pembimbing I
Ir. Muchammad Husni, M.Kom

Dosen Pembimbing II
Henning Titi Ciptaningtyas, S.Kom., M.Kom

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

AUTOMATIC CLOTH RETRIEVAL SYSTEM USING MOTOR AND SENSOR BASED ON ARDUINO

ADETIYA BAGUS NUSANTARA
NRP 05111440000004

Supervisor I
Ir. Muchammad Husni, M.Kom

Supervisor II
Henning Titi Ciptaningtyas, S.Kom., M.Kom

DEPARTMENT OF INFORMATICS
Faculty of Information and Communication Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

SISTEM JEMURAN PAKAIAN OTOMATIS MENGUNAKAN MOTOR DAN SENSOR BERBASIS ARDUINO

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Arsitektur dan Jaringan Komputer
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh :

ADETIYA BAGUS NUSANTARA

NRP : 05111440000004

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Ir. Muchammad Husni, M.Kom

NIP: 19600221 198403 1001

(pembimbing 1)

Henning Titi Ciptaningtyas, S.Kom., M.Kom

NIP: 19840708 201012 2004

(pembimbing 2)



SURABAYA

JUNI, 2018

[Halaman ini sengaja dikosongkan]

SISTEM JEMURAN PAKAIAN OTOMATIS MENGUNAKAN MOTOR DAN SENSOR BERBASIS ARDUINO

Nama Mahasiswa : Adetiya Bagus Nusantara
NRP : 05111440000004
Jurusan : Departemen Informatika FTIK-ITS
Dosen Pembimbing 1 : Ir. Muchammad Husni, M.Kom
Dosen Pembimbing 2 : Henning Titi Ciptaningtyas, S.Kom.,
M.Kom.

ABSTRAK

Tugas akhir ini adalah merancang dan membangun sebuah prototype Jemuran Pakaian Otomatis menggunakan microcontroller Arduino, motor beserta sensor-sensor yang terpasang di dalamnya. Rancang bangun prototype ini ditujukan untuk memudahkan orang ketika menjemur pakaiannya agar bila sewaktu-waktu hujan melanda, jemuran secara otomatis memindahkan pakaian menuju tempat yang teduh sehingga pakaiannya tidak basah terkena air hujan. Diharapkan dengan adanya sistem jemuran pakaian otomatis ini bisa memberi manfaat apabila diterapkan dalam kehidupan nyata.

Tantangan utama tugas akhir ini adalah menggabungkan microcontroller, beberapa sensor, motor DC dan modul wifi agar saling terkoordinir dengan baik sehingga menjadi sebuah sistem jemuran pakaian otomatis sesuai yang diharapkan.

Hasil dari Tugas Akhir ini adalah terbentuknya sistem jemuran pakaian otomatis berbasis arduino yang mana sistem ini dapat memenuhi beberapa kebutuhan fungsional antara lain dapat menggerakkan jemuran pakaian secara otomatis dalam keadaan hujan ataupun gelap, dapat memantau kondisi cuaca melalui

aplikasi smartphone serta dapat mengontrol jemuran pakaian secara manual melalui aplikasi mobile.

Kata kunci: *Arduino UNO, DHT11, Light Dependent Resistor, ESP8266, Sensor Hujan, Motor DC, Internet of Things*

AUTOMATIC CLOTH RETRIEVAL SYSTEM USING MOTOR AND SENSOR BASED ON ARDUINO

Student Name : Adetiya Bagus Nusantara
Student ID : 05111440000004
Major : Informatics Department FTIK-ITS
Supervisor I : Ir. Muchammad Husni, M.Kom
Supervisor II : Henning Titi Ciptaningtyas, S.Kom., M.Kom

ABSTRACT

This final project is to design and build a prototype of Automatic Clothes Retrieval System using Arduino microcontroller along with motor and sensors installed in it. The design of this prototype is intended to facilitate people when hanging their clothes so that if at any time the rain hit, the clothesline automatically move the clothes to the shade side so that clothes are not wet exposed to rain water. It is expected that with the automatic clothes clothesline system can provide benefits when applied in real life.

The main challenge of this final task is to combine microcontroller, some sensors, DC motor and wifi module to be well coordinated so that it becomes an automatic clothes clothesline system as expected.

The result of this Final Project is Automatic Cloth Retrieval System Based on Arduino where the system can satisfy some functional needs such as can move clothes clothes automatically in the rain or dark conditions, can monitor the weather conditions through the smartphone application and can control the clothes clothing manually through mobile apps.

Keywords: *Arduino UNO, DHT11, Light Dependent Resistor, ESP8266, Raindrop Sensor, Motor DC, Internet of Things*

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji syukur kepada Tuhan Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul:

SISTEM JEMURAN PAKAIAN OTOMATIS BERBASIS ARDUINO

Melalui lembar ini, penulis ingin menyampaikan ucapan terimakasih dan penghormatan yang sebesar-besarnya kepada:

1. Bapak, Ibu, Fauzan, David, Putri dan keluarga besar yang selalu memberikan dukungan penuh untuk menyelesaikan Tugas Akhir ini.
2. Bapak Muchammad Husni beserta keluarga dan Ibu Henning Titi Ciptaningtyas beserta keluarga selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan petunjuk selama proses pengerjaan tugas akhir ini.
3. Silvi Eka Monica yang selalu ada untuk menemani dan memberi support penulis di saat susah, sedih, dan bahagia.
4. Seluruh keluarga besar PPM Khoirul Huda I Surabaya yang selalu menemani dan membantu penulis selama menimba ilmu di pondok pesantren.
5. Rekan-rekan tim kreatif PPM Khoirul Huda Surabaya : Ayas, Sarah, Yusita, Ubaid, Nana.
6. Irzal, Juju, Kunto, Muhsin, Hendri, Amrizal, Jeffry, Kukuh dan Cahya selaku rekan yang telah menemani dan membantu pengerjaan Tugas Akhir ini.
7. Teman-teman Kenangan '14 sebagai teman seperjuangan selama di Surabaya.

8. Cak Sundarta, Cak Shobur, Cak Angga, Cak Kharis, Cak Affan sebagai teman ashabul Keputih yang membantu penulis dalam kelancaran mengerjakan Tugas Akhir.
9. Prima Nur Zaki yang telah bersedia meminjamkan laptopnya di kala laptop penulis sedang rusak.
10. Teman-teman angkatan 2014 jurusan Teknik Informatika ITS yang telah menemani dan mendukung penulis selama kuliah.
11. Google.com yang membantu penulis mengerjakan Tugas Akhir.
12. Serta pihak-pihak lain yang namanya tidak dapat penulis sebutkan satu per satu.

Bagaimanapun juga penulis telah berusaha sebaik-baiknya dalam menyusun Tugas Akhir ini, namun penulis mohon maaf apabila terdapat kekurangan yang penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Juni 2018

Adetiya Bagus Nusantara

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR KODE SUMBER	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Permasalahan.....	2
1.3 Batasan Permasalahan	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Metodologi	3
1.7 Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA.....	9
2.1 Arduino Uno.....	9
2.2 Sensor Kelembapan (<i>DHT11</i>)	10
2.3 Sensor Hujan (<i>Raindrop Sensor</i>).....	12
2.4 Sensor LDR (<i>Light Dependent Resistor</i>).....	13
2.5 <i>Motor Driver L298N</i>	14
2.6 <i>Motor Gear DC</i>	15
2.7 <i>Modul Wifi ESP8266</i>	16
2.8 <i>Saklar Microswitch</i>	17
2.9 <i>Arduino Software IDE</i>	18
2.10 <i>Logika Fuzzy</i>	19
2.11 <i>Fuzzy Logic Toolbox</i>	20
2.12 <i>eFLL (Embedded Fuzzy Logic Library)</i>	21
2.13 <i>MIT App Inventor</i>	23
2.14 <i>Thinkspeak</i> TM	24
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	25

3.1	Analisis	25
3.1.1	Analisis Permasalahan	25
3.1.1	Deskripsi Umum.....	26
3.2	Perancangan Sistem.....	26
3.2.1	Lingkungan Implementasi	27
3.2.2	Perancangan Arsitektur Jaringan	27
3.2.3	Perancangan <i>Flowchart Diagram</i> Sistem.....	28
3.2.1	Perancangan <i>Prototype</i>	29
3.2.2	Perancangan Alat.....	31
3.2.3	Kebutuhan Fungsional Sistem	32
3.2.1	Diagram Kasus Penggunaan	33
3.2.2	Perancangan Aplikasi	34
3.2.3	Perancangan Antarmuka Aplikasi	35
BAB IV IMPLEMENTASI SISTEM.....		37
4.1	Lingkungan Pengembangan Sistem.....	37
4.2	Implementasi Alat	37
4.2.1	Gambar Rangkaian alat	37
4.3	Implementasi <i>Fuzzy Logic</i>	39
4.3	Implementasi <i>Coding</i> Alat.....	49
4.3.1	Kode Sumber Pembacaan Sensor	49
4.3.2	Kode Sumber Pengiriman Data Sensor	50
4.3.2	Kode Sumber Logika Jemuran Otomatis.....	52
4.3.3	Kode Sumber Menggerakkan Jemuran ke Tempat Teduh	53
4.3.4	Kode Sumber Menggerakkan Jemuran ke <i>Normal State</i>	53
4.3.5	Kode Sumber Pengiriman Notifikasi Status Jemuran Pakaian Melalui <i>E-mail</i>	54

4.4 Implementasi Pembuatan Aplikasi <i>Mobile</i>	58
4.4.1 Kode Sumber Menyambungkan <i>API Thinkspeak</i> menuju Aplikasi <i>Mobile</i>	58
4.4.2 Kode Sumber Menampilkan Nilai Sensor pada Aplikasi <i>Mobile</i>	59
4.4.3 Hasil Pembacaan Data Sensor Suhu pada Aplikasi <i>Thinkspeak</i>	59
4.4.4 Hasil Pembacaan Data Sensor Kelembapan pada Aplikasi <i>Thinkspeak</i>	60
4.4.5 Hasil Pembacaan Data Sensor Cahaya pada Aplikasi <i>Thinkspeak</i>	60
4.4.6 Implementasi Tampilan Aplikasi <i>Mobile</i>	61
4.4.7 Kode Sumber Menggerakkan Jemuran ke Tempat Teduh Menggunakan Aplikasi <i>Mobile</i>	61
4.4.7 Kode Sumber Menggerakkan Jemuran ke <i>Normal State</i> Menggunakan Aplikasi <i>Mobile</i>	62
4.4 Implementasi Pembuatan <i>Prototype</i>	63
BAB V PENGUJIAN DAN EVALUASI	66
5.1 Skenario Pengujian.....	66
5.2 Pengujian Alat.....	66
5.3 Evaluasi Pengujian Fungsionalitas	72
BAB VI KESIMPULAN DAN SARAN.....	74
5 74	
5.1 Kesimpulan.....	74
5.2 Saran.....	75
DAFTAR PUSTAKA.....	76
BIODATA PENULIS.....	80

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1	Arduino Uno [2]	9
Gambar 2.2	Sensor Kelembapan [3]	10
Gambar 2.3	Sensor Hujan (Raindrop Sensor) [4]	12
Gambar 2.4	Sensor <i>LDR (Light Dependent Resistor)</i> [5]	13
Gambar 2.5	Motor Driver L298N [6].....	14
Gambar 2.6	<i>Motor Gear DC</i> [7]	15
Gambar 2.7	<i>Modul Wifi ESP8266</i> [6]	16
Gambar 2.8	Saklar Microswitch [7]	17
Gambar 2.9	Arduino <i>Software IDE</i>	18
Gambar 2.10	<i>Fuzzy Logic Toolbox</i> [9]	21
Gambar 2.11	Konfigurasi Dasar pada Fuzzy Logic [10].....	22
Gambar 2.12	<i>MIT App Inventor</i> [11]	23
Gambar 2.13	<i>Thinkspeak</i> [12]	24
Gambar 3.1	Rancangan Arsitektur Jaringan.....	27
Gambar 3.2	<i>Flowchart Diagram</i> Sistem	28
Gambar 3.3	Kayu [13].....	29
Gambar 3.4	Kertas Sticker [14].....	29
Gambar 3.5	<i>GT2 Pulley</i> [15].....	30
Gambar 3.6	Belt Timer [16].....	30
Gambar 3.7	Block Diagram Sistem.....	31
Gambar 3.8	<i>UseCase Diagram</i>	33
Gambar 3.9	Tampilan <i>Interface ThingSpeak</i> TM [12]	35
Gambar 3.10	Rancangan antarmuka aplikasi	35
Gambar 4.1	Rangkaian Alat	38
Gambar 4.2	Variabel <i>Input-Output Fuzzy</i>	39
Gambar 4.3	Himpunan Keanggotaan Variabel Hujan.....	41
Gambar 4.4	Himpunan Keanggotaan Variabel Cahaya.....	42
Gambar 4.5	Himpunan Keanggotaan Variabel Suhu	42
Gambar 4.6	Himpunan Keanggotaan Variabel Kelembapan	43
Gambar 4.7	Himpunan Keanggotaan Variabel Cuaca.....	43
Gambar 4.8	Tampilan <i>Fuzzy Rule Viewer</i>	46

Gambar 4.9 Hasil Implementasi Notifikasi <i>E-mail</i>	57
Gambar 4.10 Kode Sumber Menyambungkan API Thinkspeak .	58
Gambar 4.11 Kode Sumber Menampilkan Nilai Sensor	59
Gambar 4.12 Hasil Pembacaan Data Sensor Suhu	59
Gambar 4.13 Hasil Pembacaan Data Sensor Kelembapan	60
Gambar 4.14 Hasil Pembacaan Data pada Sensor Cahaya.....	60
Gambar 4.15 Implementasi Tampilan Aplikasi Mobile	61
Gambar 4.16 Kode Sumber Memasukkan Jemuran	62
Gambar 4.17 Kode Sumber Mengeluarkan Jemuran.....	63
Gambar 4.18 Implementasi Pembuatan <i>Prototype</i>	64

DAFTAR TABEL

Tabel 3.1 Lingkungan Implementasi.....	27
Tabel 3.2 Lingkungan Implementasi.....	32
Tabel 3.3 Keterangan Kode Kasus Penggunaan.....	34
Tabel 4.1 Lingkungan Pengembangan Sistem	37
Tabel 4.2 Variabel <i>Input</i>	40
Tabel 4.3 Variabel <i>Output</i>	40
Tabel 4.4 Himpunan Pembicaraan Variabel.....	41
Tabel 4.5 <i>Rules</i>	45
Tabel 5.1 Hasil Pengujian <i>Fuzzy Logic</i>	67
Tabel 5.2 Pengujian Respon Alat terhadap Sensor Hujan.....	69
Tabel 5.3 Pengujian Respon Alat terhadap Sensor Cahaya.....	70
Tabel 5.4 Pengujian Keberhasilan Aplikasi Mobile dalam Melakukan Pembacaan Data pada Sensor	71
Tabel 5.5 Pengujian Keberhasilan Sistem dalam Mengontrol Jemuran Pakaian Secara Manual Menggunakan Aplikasi Mobile	72
Tabel 5.6 Evaluasi Pengujian Fungsionalitas	73

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Inisiasi <i>Input Fuzzy</i>	48
Kode Sumber 4.2 Inisiasi <i>Output Fuzzy</i>	48
Kode Sumber 4.3 Definisi Rule <i>Fuzzy</i>	49
Kode Sumber 4.4 Pembacaan Sensor.....	50
Kode Sumber 4.5 Pengiriman Data Sensor	52
Kode Sumber 4.6 Logika Jemuran Otomatis	52
Kode Sumber 4.7 Menggerakkan Jemuran ke Tempat Teduh	53
Kode Sumber 4.8 Menggerakkan Jemuran ke <i>normal state</i>	54
Kode Sumber 4.9 Menyambungkan <i>Modul Wifi</i> menuju <i>Access Point</i>	54
Kode Sumber 4.10 Definisi E-mail Pengirim Notifikasi.....	55
Kode Sumber 4.11 Notifikasi Status Jemuran ketika Dikeluarkan	56
Kode Sumber 4.12 Notifikasi Status Jemuran ketika Dimasukkan	56
Kode Sumber 4.13 Menggerakkan Jemuran ke Tempat Teduh Menggunakan Aplikasi Mobile	62
Kode Sumber 4.14 Menggerakkan Jemuran ke Tempat Teduh Menggunakan Aplikasi Mobile	63

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

1.1 Latar Belakang

Jemuran pakaian merupakan kebutuhan yang wajib dimiliki oleh hampir setiap orang. Hujan ataupun cuaca buruk hingga saat ini menjadi masalah utama bagi masyarakat yang memiliki jemuran. Pada saat musim hujan, mayoritas orang merasa cemas ketika mereka sedang menjemur pakaian. Rasa cemas tersebut akan bertambah pada saat menjemur pakaian namun sedang berada diluar rumah dan dirumah sedang tidak ada orang. Dari kejadian itu orang jadi enggan menjemur pakain ditempat yang terbuka, karena khawatir jemurannya basah terkena air hujan.

Pemanasan global yang sekarang ini sedang terjadi menyebabkan musim di Indonesia menjadi kurang menentu, sehingga musim kemarau dan musim penghujan sudah tidak dapat diprediksikan lagi [1]. Kondisi yang tidak menentu tersebut akan sangat merepotkan apabila hendak menjemur pakaian . Rasa khawatir tersebut bertambah ketika rumah dalam keadaan kosong, sedangkan jemuran yang digunakan untuk mengeringkan pakaian basah masih berada di luar rumah sehingga pakaian yang dijemur tidak kering dengan maksimal. Dampak lebih buruknya lagi pakaian bisa menjadi lebih kotor hingga timbul bau.

Saat ini sudah banyak alat pengering pakaian yang dijual di pasaran. Namun masih banyak pebisnis *laundry* yang belum menggunakan pengering tersebut. Hal ini disebabkan oleh beberapa faktor. Salah satunya adalah harga yang relatif tinggi. Satu buah mesin pengering pakaian baju otomatis harganya bisa mencapai Rp 3.900.000,00. Ada dua jenis pengering pakaian. Ada yang berbahan bakar gas dan ada yang bertenaga listrik. Untuk tenaga gas selain membutuhkan gas untuk mengoperasikannya juga membutuhkan sekitar 300 watt tenaga listrik sedangkan untuk

tenaga listrik membutuhkan minimal 2200 watt daya listrik[12]. Hal itu tentunya membuat pebisnis *laundry* berpikir dua kali sebelum membeli mesin pengering jemuran pakaian otomatis tersebut.

Untuk mengatasi masalah tersebut dibuatlah sebuah perancangan *prototype* jemuran pakaian otomatis menggunakan motor dan sensor berbasis arduino. Jemuran pakaian otomatis ini bekerja apabila sensor LDR (*Light Dependent Resistor*), sensor hujan, dan sensor kelembapan untuk mendeteksi perubahan lingkungan sekitar. Kemudian hasil sensor tersebut diolah oleh arduino yang digunakan untuk menarik dan memasukkan jemuran menggunakan *motor gear DC*. Konsep jemuran pakaian otomatis ini nantinya dipadukan dengan konsep *Internet of Things* sehingga pengguna dapat memantau kondisi cuaca di rumah dan mengontrol kondisi jemurannya di manapun dia berada.

Dengan adanya sistem jemuran pakaian otomatis berbasis arduino ini pengguna tidak perlu mengeluarkan biaya sebesar itu. Cukup menghabiskan dana sekitar Rp 500.000,00 jemuran pakaian otomatis sudah bisa dibuat. Selain itu kekurangan dari alat pengering pakaian otomatis yang ada adalah terbatasnya pakaian yang bisa dijemur. Hanya bisa menampung sekitar 20 pakaian karena terbatasnya ruang, sementara jemuran pakaian otomatis ini bisa menampung banyak pakaian tergantung panjang rel jemuran yang ingin digunakan.

1.2 Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini antara lain:

1. Bagaimana cara merancang Sistem Jemuran Pakaian Otomatis Menggunakan Motor dan Sensor Berbasis Arduino?
2. Bagaimana cara membuat aplikasi *mobile* untuk memantau kondisi jemuran di rumah?
3. Bagaimana cara membuat aplikasi *mobile* untuk mengontrol jemuran pakaian secara manual?

1.3 Batasan Permasalahan

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, antara lain:

1. Jemuran pakaian otomatis dibuat menggunakan *Microcontroller* Arduino
2. Rancang bangun jemuran pakaian otomatis ini adalah simulasi sehingga belum bisa dipakai untuk menahan berat jemuran pakaian yang sesungguhnya.
3. Penggerak jemuran yang digunakan adalah *Motor Gear DC*.
4. Aplikasi yang dibuat berbasis perangkat bergerak untuk telepon pintar Android dengan versi minimal 4.1 *Jelly Bean*.

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah:

1. Membuat jemuran pakaian otomatis menggunakan motor dan sensor berbasis arduino
2. Membangun aplikasi *mobile* untuk memantau kondisi cuaca di rumah.
3. Membangun aplikasi *mobile* untuk mengontrol jemuran secara manual.

1.5 Manfaat

Dengan dibuatnya tugas akhir ini diharapkan bisa memberi manfaat yang besar dan mengurangi rasa khawatir bagi pengguna ketika menjemur pakaiannya.

1.6 Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

a. Penyusunan proposal tugas akhir

Proposal tugas akhir ini berisi latar belakang pembuatan tugas akhir, rumusan masalah, batasan masalah, tujuan pembuatan, manfaat, metodologi hingga jadwal kegiatan pembuatan tugas akhir. Selain itu proposal tugas akhir ini memberikan ringkasan dari tugas akhir. Proposal tugas akhir juga berisi tinjauan pustaka yang digunakan sebagai referensi pembuatan tugas akhir ini.

b. Studi literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan program yaitu mengenai *Internet of Things (IoT)*, *Fuzzy Logic*, Arduino, sensor, *wifi module* dan Android.

c. Analisis dan desain perangkat lunak

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dibuat *prototype* sistem, yang merupakan rancangan dari sistem jemuran pakaian otomatis berbasis arduino beserta rancangan aplikasi berbasis android sebagai penunjangnya.

d. Implementasi perangkat keras

Implementasi perangkat keras dibangun menggunakan Arduino sebagai *microcontrollernya* disertai dengan beberapa sensor yang terpasang yaitu sensor kelembapan *DHT11*, sensor hujan (*raindrop sensor*) dan sensor LDR (*Light Dependent Resistor*). Adapun jemuran digerakkan menggunakan *motor gear DC* dan data dikirimkan menggunakan *modul wifi* ESP8266.

e. Pengujian dan evaluasi

Pada tahapan ini dilakukan uji coba terhadap perangkat keras dan perangkat lunak yang telah dibuat. Pengujian dan evaluasi akan dilakukan dengan melihat kesesuaian dengan perencanaan. Tahap ini dimaksudkan juga untuk mengevaluasi jalannya sistem, mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

f. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
 - a. Latar Belakang
 - b. Rumusan Masalah
 - c. Batasan Tugas Akhir
 - d. Tujuan
 - e. Metodologi
 - f. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

1.7 Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Dasar Teori

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses dan perancangan antarmuka aplikasi.

Bab IV Implementasi

Bab ini berisi implementasi dari perancangan dan implementasi fitur-fitur penunjang aplikasi.

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dengan metode kotak hitam (*black box testing*) untuk mengetahui aspek nilai fungsionalitas dari perangkat lunak dan nilai kegunaan yang dibuat dengan juga memperhatikan ketertarikan pada calon partisipan untuk menggunakan aplikasi ini.

Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

Lampiran

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

[Halaman ini sengaja dikosongkan]

BAB II TINJAUAN PUSTAKA

Pada bab ini akan dipaparkan mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir ini, di antaranya adalah Arduino Uno, Sensor Kelembapan (*DHT11*), Sensor Hujan (*Raindrop Sensor*), Sensor Cahaya, Logika *Fuzzy* dan beberapa teori lainnya yang menunjang dalam pengerjaan penelitian.

2.1 Arduino Uno



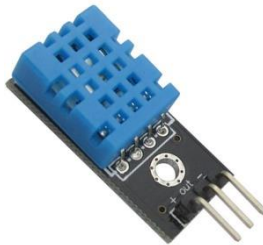
Gambar 2.1 Arduino Uno [2]

Arduino Uno adalah board mikrokontroler berbasis ATmega328 (*datasheet*). Memiliki 14 *pin input* dari *output digital* dimana 6 *pin input* tersebut dapat digunakan sebagai *output PWM* dan 6 *pin input analog*, 16 MHz *osilator kristal*, koneksi USB, *jack power*, *ICSP header*, dan tombol *reset*. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan *Board* Arduino Uno ke komputer dengan menggunakan kabel USB atau listrik dengan AC ke *adaptor-DC* atau baterai untuk menjalankannya. [2]

Untuk membangun Sistem Jemuran Pakaian Otomatis ini tentunya diperlukan sebuah microcontroller untuk mengatur jalannya sistem agar saling terkoordinir dengan baik.

Dalam pengerjaan Tugas Akhir kali ini Arduino Uno dipilih sebagai *microkontroller* dikarenakan *compatible* dengan beberapa komponen yang nantinya digunakan. Selain itu arduino harganya relatif lebih murah dan didukung dengan ratusan *library* yang ada. Pemrograman arduino tergolong mudah digunakan dan termasuk *Open-Source* baik dari segi *software* maupun dari segi *hardware*.

2.2 Sensor Kelembapan (DHT11)



Gambar 2.2 Sensor Kelembapan [3]

DHT11 adalah sensor Suhu dan Kelembapan. Sensor ini memiliki keluaran sinyal digital yang dikalibrasi dengan sensor suhu dan kelembapan yang kompleks. Teknologi ini memastikan keandalan tinggi dan sangat baik stabilitasnya dalam jangka panjang. mikrokontroler terhubung pada kinerja tinggi sebesar 8 bit. Sensor ini termasuk elemen resistif dan perangkat pengukur suhu *NTC*. Memiliki kualitas yang sangat baik, respon cepat, kemampuan anti-gangguan dan keuntungan biaya tinggi kinerja. Pada prinsipnya cara kerja sensor ini adalah mendeteksi besarnya kelembapan relatif udara di sekitar sensor tersebut. [3]

DHT11 yang mendeteksi kelembapan di sekitarnya akan merubah frekuensi *oscillator* dan akan mengirimkan data ke

mikrokontroler *slave*. Dari mikrokontroller *slave* akan dilanjutkan ke mikrokontroller *master*. Selanjutnya mikrokontroller akan menganalisa data, mikrokontroller melakukan dengan cara membandingkan antara data yang dikirim dan data masukan. Apabila dalam membandingkan tersebut diatas kelembapan yang ditentukan dibawah atau diatas dari data yang dikirim sensor maka alat akan bekerja untuk menyesuaikan kelembapan menjadi sesuai dengan yang diharapkan. [3]

Untuk membangun Sistem Jemuran Pakaian Otomatis ini diperlukan beberapa sensor untuk mendeteksi kondisi cuaca dari lingkungan sekitar. Salah satu sensor yang diperlukan adalah sensor kelembapan.

Dalam pengerjaan Tugas Akhir kali ini sensor kelembapan yang digunakan adalah sensor DHT11. Sensor DHT11 dipilih karena memiliki fitur kalibrasi sangat akurat dari kelembapan ruang kalibrasi. Koefisien kalibrasi yang disimpan dalam memori program OTP, sensor internal mendeteksi sinyal dalam proses, kita harus menyebutnya koefisien kalibrasi. Sistem antarmuka tunggal-kabel serial terintegrasi untuk menjadi cepat dan mudah. Kecil ukuran, daya rendah, sinyal transmisi jarak hingga 20 meter, sehingga berbagai aplikasi dan bahkan aplikasi yang paling menuntut. Produk ini 4-pin pin baris paket tunggal. Koneksi nyaman, paket khusus dapat diberikan sesuai dengan kebutuhan pengguna.

2.3 Sensor Hujan (*Raindrop Sensor*)



Gambar 2.3 Sensor Hujan (Raindrop Sensor) [4]

Sensor hujan adalah sensor yang berfungsi sebagai pendeteksi ada tidaknya kondisi rintik hujan, yang dapat dimanfaatkan dalam berbagai aplikasi. Di pasaran sensor ini dijual dalam bentuk modul sehingga hanya perlu menyediakan kabel *jumper* untuk dihubungkan ke mikrokontroler atau Arduino. Cara kerja dari sensor air hujan adalah ketika sensor terkena air hujan maka jalur *port* dan jalur *ground* terhubung sehingga tidak ada tegangan karena *port* langsung terhubung langsung dengan *ground*. Sensor hujan berfungsi untuk memberikan nilai masukan pada tingkat elektrolisis air hujan, dimana panel sensor hujan akan tersentuh oleh air hujan yang turun. [4]

Untuk membangun Sistem Jemuran Pakaian Otomatis ini diperlukan beberapa sensor untuk mendeteksi kondisi cuaca dari lingkungan sekitar. Salah satu sensor yang diperlukan adalah sensor hujan karena hujan merupakan masalah terbesar ketika menjemur pakaian.

Dalam pengerjaan Tugas Akhir kali ini sensor hujan dipilih karena sensor ini terbilang murah dan *compatible* dengan *microcontroller* arduino.

2.4 Sensor LDR (*Light Dependent Resistor*)



Gambar 2.4 Sensor LDR (*Light Dependent Resistor*) [5]

Light Dependent Resistor atau disingkat dengan LDR adalah jenis Resistor yang nilai hambatan atau nilai resistansinya tergantung pada intensitas cahaya yang diterimanya. Nilai Hambatan LDR akan menurun pada saat cahaya terang dan nilai Hambatannya akan menjadi tinggi jika dalam kondisi gelap. Dengan kata lain, fungsi LDR (*Light Dependent Resistor*) adalah untuk menghantarkan arus listrik jika menerima sejumlah intensitas cahaya (Kondisi Terang) dan menghambat arus listrik dalam kondisi gelap. LDR Adalah jenis resistor yang nilai resistansinya dapat berubah-ubah hampir sama dengan potensiometer .Yang membedakan antara LDR dan potensiometer adalah nilai resistansi LDR berubah akibat perubahan intensitas cahaya yang diterimanya sedangkan pada potensiometer hambatan berubah akibat putaran. Jika di lihat dari fisiknya bentuk fisik ldr mempunyai 2 kaki dengan sebuah sensor di permukaannya yang peka terhadap perubahan intensitas cahaya. [5]

Untuk membangun Sistem Jemuran Pakaian Otomatis ini diperlukan beberapa sensor untuk mendeteksi kondisi dari lingkungan sekitar. Salah satu sensor yang diperlukan adalah sensor cahaya. Sensor cahaya digunakan untuk mendeteksi apakah kondisi gelap atau terang dikarenakan terkadang hujan bisa muncul secara tiba-tiba. Untuk mengantisipasi hal tersebut maka sensor cahaya digunakan agar ketika cuaca mendung maka sistem dapat menggerakkan jemuran ke tempat teduh.

Dalam pengerjaan Tugas Akhir kali ini sensor cahaya dipilih karena sensor ini terbilang murah dan *compatible* dengan *microcontroller* arduino.

2.5 *Motor Driver L298N*



Gambar 2.5 Motor Driver L298N [6]

Motor Gear DC tidak dapat dikendalikan secara langsung oleh mikrokontroler, karena kebutuhan arus yang besar sedangkan keluaran arus dari mikrokontroler sangat kecil. *Motor driver* merupakan alternatif yang dapat digunakan untuk menggerakkan *Motor Gear DC*.

Untuk membangun Sistem Jemuran Pakaian Otomatis ini diperlukan adanya motor untuk menggerakkan jemuran pakaian. Kali ini jenis motor yang digunakan adalah motor DC. Seperti penjelasan sebelumnya bahwa *Motor Gear DC* tidak dapat dikendalikan langsung arah perputarannya dan kecepatan putarannya. Maka diperlukanlah *motor driver L298N*.

Dalam pengerjaan Tugas Akhir kali ini *motor driver* yang digunakan adalah jenis L298N karena banyak dijual di pasaran dan mudah untuk digunakan. Komponen ini juga dapat mengontrol dua motor DC sekaligus. Tentunya dengan hal ini bisa memberi prospek untuk mengembangkan sistem yang lain.

2.6 *Motor Gear DC*



Gambar 2.6 *Motor Gear DC* [7]

Motor Gear DC adalah motor listrik yang memerlukan suplai tegangan arus searah pada kumparan medan untuk diubah menjadi energi gerak mekanik. Kumparan medan pada *Motor Gear DC* disebut *stator* (bagian yang tidak berputar) dan kumparan jangkar disebut *rotor* (bagian yang berputar). Motor arus searah, sebagaimana namanya, menggunakan arus langsung yang tidak langsung / *direct-undirectional*. *Motor Gear DC* memiliki manfaat yang sangat banyak dalam kehidupan sehari-hari dan dalam dunia industri. *Motor Gear DC* memudahkan pekerjaan sehingga proses industri dapat berjalan efisien. Semakin banyak industri yang berkembang, maka akan semakin banyak mesin yang digunakan. Semakin banyak mesin yang digunakan, maka semakin banyak penggunaan *Motor Gear DC*. [7]

Untuk membangun Sistem Jemuran Pakaian Otomatis ini diperlukan adanya motor untuk menggerakkan jemuran pakaian. Kali ini jenis motor yang digunakan adalah *Motor Gear DC*.

Rel jemuran digerakan dengan menggunakan *motor gear DC* yang dipasang pada ujung rel jemuran. Pemilihan *motor gear DC* didasarkan pada putaran dan torsi yang lebih besar

dibandingkan dengan *motor stepper* atau *motor servo*, juga didasarkan atas ketersediaan di pasaran selain harga murah juga banyak variasinya.

2.7 Modul Wifi ESP8266



Gambar 2.7 Modul Wifi ESP8266 [6]

ESP8266 merupakan *modul wifi* yang berfungsi sebagai perangkat tambahan *microcontroller* seperti Arduino agar dapat terhubung langsung dengan *wifi* dan membuat koneksi TCP/IP. Modul ini membutuhkan daya sekitar 3.3v dengan memiliki tiga mode wifi yaitu *Station*, *Access Point* dan *Both* (Keduanya). Modul ini juga dilengkapi dengan prosesor, memori dan GPIO dimana jumlah pin bergantung dengan jenis ESP8266 yang kita gunakan. [6]

Modul ESP8266 merupakan *platform* yang sangat murah tetapi benar-benar efektif untuk digunakan berkomunikasi atau kontrol melalui internet baik digunakan secara *standalone* (berdiri sendiri) maupun dengan menggunakan mikrokontroler tambahan dalam hal ini Arduino sebagai pengendalinya.

Untuk membangun Sistem Jemuran Pakaian Otomatis ini diperlukan adanya *modul wifi* untuk mengirimkan data sensor menuju *server* dan mengontrol jemuran pakaian. Maka digunakan *modul wifi*.

Dalam pengerjaan Tugas Akhir kali ini *modul wifi* yang digunakan adalah jenis modul wifi ESP8266 karena merupakan

modul wifi yang compatible dengan microcontroller arduino dan harganya relatif lebih murah bila dibandingkan dengan menggunakan Arduino Ethernet Shield.

2.8 Saklar *Microswitch*



Gambar 2.8 Saklar Microswitch [7]

Istilah mikro pada saklar mikro atau micro switch, tidak berarti bahwa saklar ini sendiri berukuran kecil. Nama ini mengindikasikan bahwa tombol yang digunakan untuk mengoperasikan saklar mikro, hanya bergeser dengan jarak perpindahan yang sangat kecil.

Saklar jenis ini sangat sensitif, sedikit tekanan saja pada tuas dapat mengakibatkan saklar berpindah dari satu posisi ke posisi lainnya. Kebanyakan micro switch memiliki kontak-kontak jenis SPDT, sehingga saklar ini dapat digunakan untuk menyambungkan atau memutuskan, atau keduanya secara bersamaan. [7]

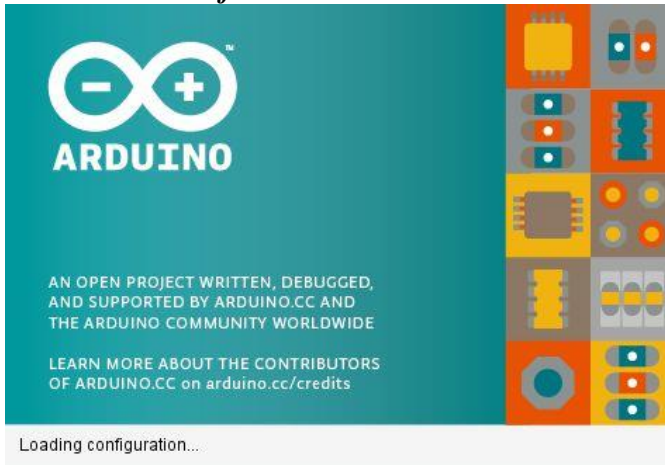
Kontak-kontak SPDT pada micro switch umumnya terdiri dari tiga buah kaki atau tag terminal, yaitu Common atau jalur bersama, NO (*Normally Open*) atau kontak normal terbuka, dan NC (*Normally Closed*) atau kontak normal tertutup. Kontak-kontak tersebut dilengkapi dengan pegas, dalam keadaan normal, kontak jalur bersama akan tersambung ke kontak normal tertutup.

Saklar *microswitch* merupakan salah satu jenis saklar jenis *pushbutton* yang mempunyai sensitifitas cukup tinggi. Dalam pembuatan alat ini saklar *microswitch* digunakan sebagai penyambung inputan *ground* pada rangkaian untuk memberikan perintah *stop* pada putaran motor.

Untuk membangun Sistem Jemuran Pakaian Otomatis ini diperlukan adanya saklar untuk menghentikan pergerakan motor.

Dalam pengerjaan Tugas Akhir kali ini saklar yang digunakan adalah saklar *microswitch*. Saklar *microswitch* ini tergolong murah dan *compatible* dengan arduino.

2.9 Arduino Software IDE



Gambar 2.9 Arduino Software IDE

IDE merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui software inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, *IC microcontroller* Arduino telah ditanamkan suatu program bernama *Bootloader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler.

Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan *library C/C++* yang biasa disebut *Wiring* yang membuat operasi input dan output menjadi lebih mudah. Arduino IDE ini dikembangkan dari software Processing yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino.

Arduino IDE digunakan untuk mengembangkan pembuatan fungsional alat pada sistem jemuran pakaian otomatis ini. Versi yang digunakan untuk mengembangkan sistem ini adalah Arduino IDE versi 1.8.5. IDE ini tergolong ringan dan mudah untuk dipelajari.

2.10 Logika Fuzzy

Konsep *Fuzzy Logic* diperkenalkan oleh Prof. Lotfi Zadeh dari Universitas California di Berkeley pada tahun 1965. Lotfi Zadeh mengemukakan bahwa *fuzzy* merupakan suatu aspek ketidakpastian yang berbeda dengan keacakan. Kemudian Zadeh mengusulkan bentuk matematika untuk melihat bagaimana ketidakjelasan dapat dinyatakan dalam bahasa manusia yang pendekatannya disebut “*logika fuzzy*”. [8]

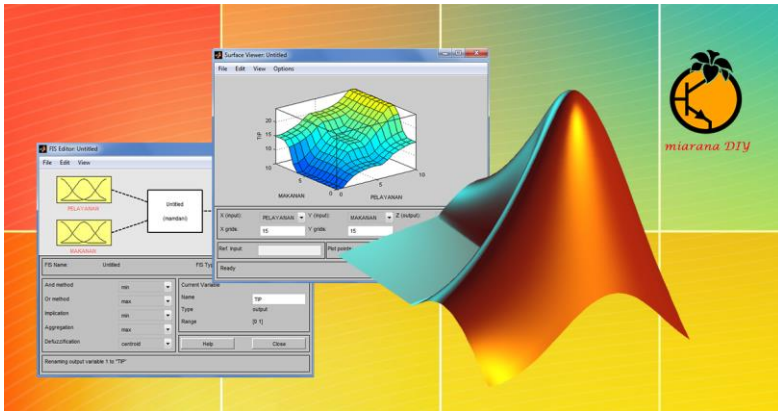
Konsep *Fuzzy Logic* umumnya diterapkan pada masalah-masalah yang mengandung unsur ketidakpastian (*uncertainty*), ketidaktepatan (*imprecise*), *noisy*, dan sebagainya. *Fuzzy logic* menjembatani bahasa mesin yang presisi dengan bahasa manusia yang menekankan pada makna atau arti (*significance*). *Fuzzy logic* dikembangkan berdasarkan cara berfikir manusia. Dalam *fuzzy* dikenal derajat keanggotaan yang memiliki interval $[0,1]$. Dalam teori logika *fuzzy* suatu nilai bisa bernilai benar atau salah secara bersama, namun berapa besar keberadaan dan kesalahan tergantung pada bobot keanggotaan yang dimilikinya.

Ada tiga proses utama jika ingin mengimplementasikan *fuzzy logic* pada suatu perangkat, yaitu fuzzifikasi, evaluasi *rule* dan defuzzifikasi. [8]

1. ***Fuzzification***, merupakan suatu proses untuk mengubah suatu masukan dari bentuk tegas(*crisp*) menjadi *fuzzy* yang biasanya disajikan dalam bentuk himpunan-himpunan *fuzzy* dengan suatu fungsi kenggotaannya masing-masing.
2. ***Interference System (Evaluasi Rule)***, merupakan sebagai acuan untuk menjelaskan hubungan antara variabel-variabel masukan dan keluaran yang mana variabel yang diproses dan yang dihasilkan berbentuk *fuzzy*. Untuk menjelaskan hubungan antara masukan dan keluaran biasanya menggunakan “IF-THEN”.
3. ***Defuzzification***, merupakan proses pengubahan variabel berbentuk *fuzzy* tersebut menjadi data-data pasti (*crisp*) yang dapat dikirimkan ke peralatan pengendalian.

Dalam pengerjaan Tugas Akhir kali ini metode *Fuzzy Logic* digunakan untuk menentukan langkah dalam pengambilan keputusan. Dengan menggunakan perhitungan *fuzzy logic* diharapkan hasil yang diberikan lebih akurat dan optimal. Beberapa keuntungan dari *fuzzy logic* antara lain : performa, kesederhaan, biaya rendah dan produktifitasnya.

2.11 *Fuzzy Logic Toolbox*



Gambar 2.10 Fuzzy Logic Toolbox [9]

MATLAB adalah bahasa pemrograman teknis yang sangat andal untuk proses simulasi dan visualisasi data. MATLAB menjadi tools bagi para ilmuwan dan pelajar untuk memodelkan sistem, menganalisis serta menampilkan data. Saat ini, MATLAB terus berkembang untuk menyokong berbagai disiplin keilmuan, termasuk Logika *Fuzzy*. MATLAB telah menyediakan sebuah tools untuk merancang logika *fuzzy*, yang dikenal sebagai *Fuzzy Logic Toolbox*. [9]

Dalam pengerjaan Tugas Akhir kali ini metode *Fuzzy Logic Toolbox* digunakan untuk mendesain logika *fuzzy* yang nantinya diterapkan pada sistem sebelum diimplementasikan pada *microcontroller* Arduino.

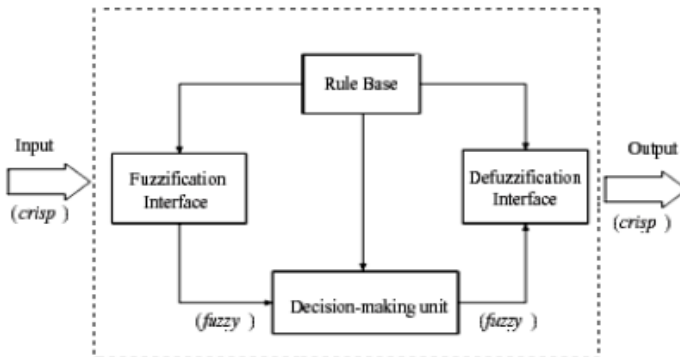
Versi Matlab yang digunakan pada pengerjaan Tugas Akhir kali ini adalah versi R2015a, sedangkan versi *Fuzzy Logic Toolbox* yang digunakan adalah versi 2.2.21

2.12 eFLL (Embedded Fuzzy Logic Library)

eFLL (Embedded Fuzzy Logic Library) dikembangkan oleh *Robotic Research Group (RRG)* di Universitas Negeri Piauí (UESPI-Teresina). *eFLL (Embedded Fuzzy Logic Library)* adalah pilihan yang fleksibel, ringan dan efisien untuk bekerja dengan *Fuzzy Logic* dalam sistem *embedded*. [10]

Dalam pengerjaan Tugas Akhir kali ini metode *eFLL (Embedded Fuzzy Logic Library)* digunakan untuk mengimplementasikan logika *fuzzy system* pada *microcontroller* Arduino.

Versi *eFLL* yang digunakan pada pengerjaan Tugas Akhir kali ini adalah v1.0.10 [*Updated at: 21 August 2016*].



Gambar 2.11 Konfigurasi Dasar pada Fuzzy Logic [10]

2.13 MIT App Inventor



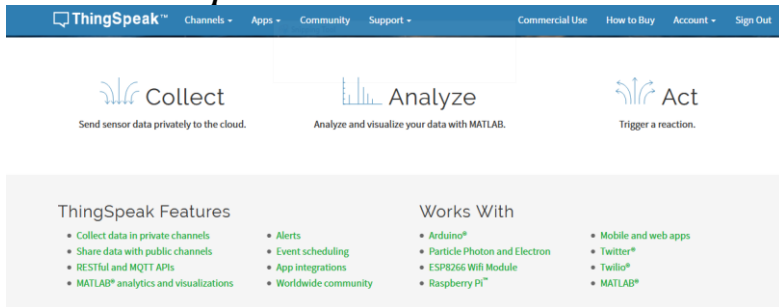
Gambar 2.12 MIT App Inventor [11]

App Inventor adalah aplikasi *web* sumber terbuka yang awalnya dikembangkan oleh Google, dan saat ini dikelola oleh Massachusetts Institute of Technology (MIT). *App Inventor* memungkinkan pengguna baru untuk memprogram komputer untuk menciptakan aplikasi perangkat lunak bagi sistem operasi Android. *App Inventor* menggunakan antarmuka grafis, serupa dengan antarmuka pengguna pada *Scratch* dan *StarLogo TNG*, yang memungkinkan pengguna untuk *men-drag-and-drop* objek visual untuk menciptakan aplikasi yang bisa dijalankan pada perangkat Android. Dalam menciptakan *App Inventor*, Google telah melakukan riset yang berhubungan dengan komputasi edukasional dan menyelesaikan lingkungan pengembangan online Google. [11]

Untuk membangun Sistem Jemuran Pakaian Otomatis ini diperlukan adanya IDE untuk membangun aplikasi *mobile* pada android. Salah satu aplikasi yang bisa digunakan MIT *App Inventor*.

Dalam pengerjaan Tugas Akhir kali ini versi *MIT App Inventor* yang digunakan adalah versi 2 di mana dengan menggunakan aplikasi ini kita bisa mengambil data langsung dari *web server* yang digunakan.

2.14 *Thinkspeak*™



Gambar 2.13 *Thinkspeak* [12]

ThingSpeak™ adalah layanan *platform* analisis IoT berbasis *cloud* yang memungkinkan penggunaanya untuk mengumpulkan, memvisualisasikan dan menganalisis aliran data. *ThingSpeak* memberikan visualisasi instan terhadap data yang diposting oleh perangkat pengguna.

Untuk membangun Sistem Jemuran Pakaian Otomatis ini diperlukan adanya *server* untuk mengelola data yang dikirimkan sensor melalui *modul wifi*. Untuk mengatasi hal tersebut digunakan *Thinkspeak*. *Thinkspeak* merupakan layanan *cloud* gratis yang bisa digunakan bebas..

Dalam pengerjaan tugas akhir kali ini *Thinkspeak* digunakan karena memiliki beberapa fitur diantaranya mengumpulkan data dalam chanel privat, mendukung Restful dan MQTT API, analisis dan visualisasi berbasis MATLAB®, mendukung *Alert*, *event scheduling*, integrasi App dan dukungan komunitas global. Thingspeak dapat bekerja pada perangkat Arduino, *Particle Photon* dan Elektron, *WiFi modul ESP8266* dan Raspberry Pi. Thingspeak juga mendukung integrasi pada aplikasi *mobile* dan *web*, Twitter, Twillio dan Matlab.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tahap analisis dan perancangan sistem yang akan dibangun. Analisis membahas semua persiapan yang akan menjadi pokok pikiran pembuatan jemuran pakaian otomatis berbasis arduino ini. Mulai dari masalah yang melatarbelakangi, hingga analisis gambaran awal sistem yang akan dibuat. Perancangan sistem membahas hal-hal yang berkaitan dengan pondasi atau dasar pembuatan alat dan aplikasi, yang meliputi perancangan alat, tampilan antar muka halaman aplikasi, hingga perancangan alur proses yang akan diimplementasikan di dalam aplikasi.

3.1 Analisis

Tahap analisis meliputi analisis masalah, analisis kebutuhan, deskripsi umum sistem, dan kasus penggunaan sistem yang dibuat.

3.1.1 Analisis Permasalahan

Jemuran pakaian merupakan kebutuhan yang wajib dimiliki oleh hampir setiap orang. Hujan ataupun cuaca buruk hingga saat ini menjadi masalah utama bagi masyarakat yang memiliki jemuran. Pada saat musim hujan, mayoritas orang merasa cemas ketika mereka sedang menjemur pakaian. Rasa cemas tersebut akan bertambah pada saat menjemur pakaian namun sedang berada diluar rumah dan dirumah sedang tidak ada orang. Dari kejadian itu orang jadi enggan menjemur pakain ditempat yang terbuka, karena khawatir jemurannya basah terkena air hujan. Pemanasan global yang sekarang ini sedang terjadi menyebabkan musim di Indonesia menjadi kurang menentu, sehingga musim kemarau dan musim penghujan sudah tidak dapat diprediksikan lagi. Rasa khawatir tersebut bertambah ketika rumah dalam keadaan kosong, sedangkan jemuran yang digunakan untuk mengeringkan pakaian

basah masih berada di luar rumah sehingga pakaian yang dijemur tidak kering dengan maksimal dan yang lebih buruknya lagi dapat menjadi lebih kotor hingga timbulnya bau.

3.1.1 Deskripsi Umum

Berdasarkan permasalahan yang ada pada pembahasan domain permasalahan, solusi yang ditawarkan adalah pembuatan jemuran pakaian otomatis menggunakan motor dan sensor berbasis arduino. Jemuran ini dapat secara otomatis memindahkan jemuran dari tempat yang tidak teduh menuju ke tempat yang teduh. Dalam pembuatan rancangan ini akan digunakan beberapa sensor yaitu sensor hujan, sensor kelembapan *DHT11* dan sensor LDR (*Light Dependent Resistor*). Apabila sensor hujan tertetes air, maka jemuran otomatis berpindah ke tempat yang teduh dan bila sudah kering atau tidak hujan maka jemuran berpindah ke tempat yang tidak teduh lagi. Begitu pula ketika gelap, jemuran otomatis berpindah ke tempat yang teduh dan bila terang maka jemuran berpindah ke tempat yang tidak teduh lagi. Selain itu dibuat sebuah aplikasi berbasis Android yang dapat memantau kondisi suhu, kelembapan, intensitas hujan dan intensitas cahaya secara *realtime*. Selain itu di dalam aplikasi terdapat tombol untuk mengontrol jemuran secara manual agar pengguna dapat menggiring jemuran ke arah yang teduh di saat diperlukan.

3.2 Perancangan Sistem

Perancangan sistem dilakukan untuk menggambarkan rancangan dan arsitektur yang akan digunakan untuk penelitian. Juga menggambarkan bagaimana proses-proses yang harus dijalani. Proses-proses ini akan dipresentasikan dengan diagram alir (*flowchart*) pada setiap uji coba yang dilakukan.

3.2.1 Lingkungan Implementasi

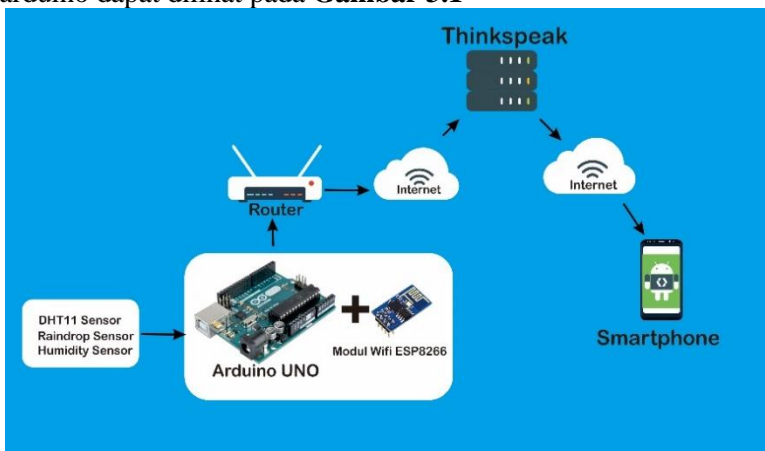
Lingkungan implementasi untuk pengerjaan tugas akhir ini adalah ditunjukkan pada **Tabel 3.1** berikut:

Tabel 3.1 Lingkungan Implementasi

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat keras	Prosesor	Intel® Core™ i5-3337U (1.80GHz)
	RAM	8 GB
	System Type	64-bit <i>Operating System</i>
Perangkat lunak	Sistem Operasi	Windows 10 Enterprise N
	IDE	Arduino <i>Software IDE</i> , <i>MIT App Inventor</i>

3.2.2 Perancangan Arsitektur Jaringan

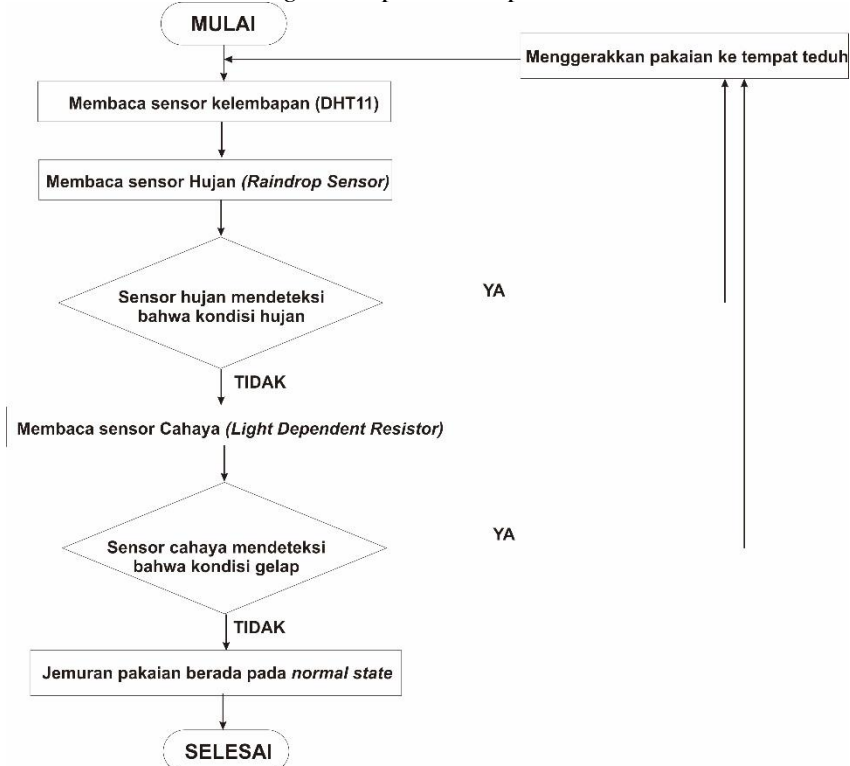
Berikut ini adalah rancangan arsitektur jaringan yang akan dibuat dalam sistem jemuran pakaian otomatis berbasis arduino dapat dilihat pada **Gambar 3.1**



Gambar 3.1 Rancangan Arsitektur Jaringan

3.2.3 Perancangan *Flowchart Diagram* Sistem

Pada tahap ini akan dijelaskan perancangan *flowchart diagram* cara kerja sistem jemuran pakaian otomatis berbasis arduino. *Flowchart Diagram* dapat dilihat pada **Gambar 3.2**.



Gambar 3.2 *Flowchart Diagram* Sistem

3.2.1 Perancangan *Prototype*

Pada tahap ini akan dijelaskan perancangan *prototype* untuk membangun jemuran pakaian otomatis berbasis arduino . Dalam membangun *prototype* ini digunakan beberapa bahan yang diperlukan yaitu :

a. Kayu



Gambar 3.3 Kayu [13]

Kayu digunakan untuk membuat *prototype* jemuran pakaian otomatis ini dikarenakan kokoh dan kuat bila dibandingkan dengan kertas *duplex*, kertas karton ataupun kardus. Selain itu kayu juga mudah didapatkan dan mudah untuk dipotong sesuai dengan keinginan.

b. Kertas *Sticker*



Gambar 3.4 Kertas Sticker [14]

Kertas *sticker* dipilih untuk membuat *prototype* jemuran pakaian otomatis ini dikarenakan selain harganya lebih terjangkau bila dibandingkan dengan cat hasilnya juga lebih rapi bila dibandingkan dengan mengecat. Untuk mendesain *sticker* sendiri juga tergolong mudah. Dengan menggunakan aplikasi semisal *Corel Draw* kita bisa membuat rancangan sendiri dengan mudah.

c. *GT2 Pulley*



Gambar 3.5 GT2 Pulley [15]

GT2 Pulley digunakan untuk menggerakkan jemuran pada sistem jemuran. Alat ini harganya cukup terjangkau dan bisa dibeli dengan mudah melalui toko *online*.

d. *Belt Timer*



Gambar 3.6 Belt Timer [16]

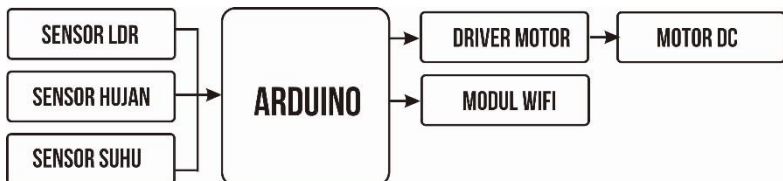
Belt Timer digunakan untuk menggerakkan jemuran pada sistem jemuran. Alat ini merupakan pasangan dari *GT2 Pulley*. Alat ini harganya cukup terjangkau dan bisa dibeli dengan mudah melalui toko *online*.

3.2.2 Perancangan Alat

Pada tahap ini akan dijelaskan perancangan alat untuk membangun jemuran pakaian otomatis menggunakan motor dan sensor berbasis arduino . Dalam membangun alat ini digunakan beberapa komponen yang diperlukan yaitu :

- e. Arduino Uno
- f. Papan *Breadboard*
- g. Kabel *Jumper*
- h. Sensor Hujan (*Raindrop Sensor*)
- i. Sensor Kelembapan DHT11
- j. Sensor LDR(*Light Dependent Resistor*)
- k. *Modul Wifi* ESP8266
- l. Saklar *Microswitch*
- m. *Motor Gear DC*
- n. *Driver Motor L298N*

Berikut ini adalah gambar diagram blok rangkaian komponen pada sistem jemuran pakaian otomatis berbasis arduino dapat dilihat pada **Gambar 3.7**.



Gambar 3.7 Block Diagram Sistem

3.2.3 Kebutuhan Fungsional Sistem

Kebutuhan fungsional berisikan proses-proses yang dibutuhkan dalam sistem dan harus dijalankan. Adapun kebutuhan fungsional sistem jemuran pakaian otomatis ini dideskripsikan dalam **Tabel 3.2**

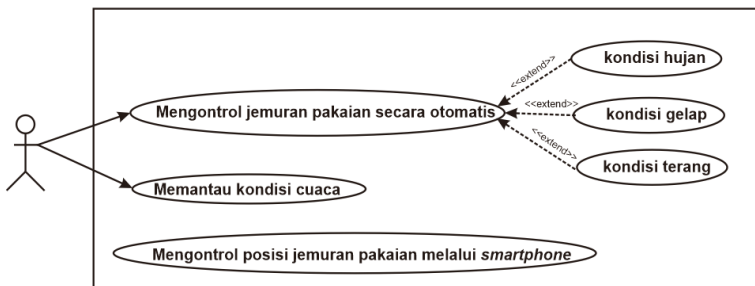
Tabel 3.2 Lingkungan Implementasi

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
F-001	<i>Mengontrol jemuran pakaian secara otomatis ketika kondisi hujan</i>	Sistem dapat menggerakkan jemuran pakaian secara otomatis apabila terjadi hujan atau kondisi terang.
F-002	<i>Mengontrol jemuran pakaian secara otomatis ketika kondisi gelap</i>	Sistem dapat menggerakkan jemuran pakaian secara otomatis apabila terjadi kondisi gelap atau kondisi terang.
F-003	Memantau kondisi cuaca	Pengguna dapat memantau kondisi cuaca terkait dengan tingkat curah hujan, intensitas kelembapan, intensitas suhu, tingkat kecerahan melalui aplikasi yang terpasang pada <i>smartphone</i>

F-004	Mengontrol jemuran pakaian secara manual melalui <i>Smartphone</i>	Pengguna dapat menggerakkan jemuran pakaian secara otomatis melalui aplikasi yang terpasang pada <i>smartphone</i>
--------------	--	--

3.2.1 Diagram Kasus Penggunaan

Kasus penggunaan yang dibutuhkan pada sistem sesuai dengan analisa yang telah dilakukan. Berikut adalah diagram kasus penggunaan dari sistem jemuran pakaian otomatis berbasis arduino dapat dilihat di **Gambar 3.8**



Gambar 3.8 UseCase Diagram

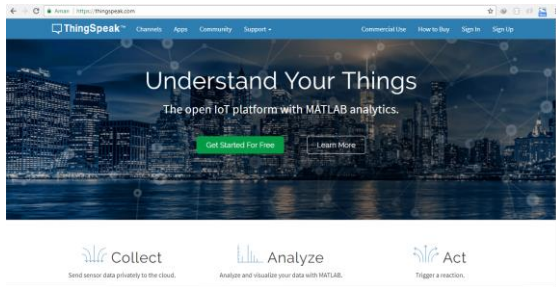
Dari gambar diagram kasus penggunaan yang ada pada **Gambar 3.8** keterangan kode kasus penggunaan dapat dilihat pada **Tabel 3.3**

Tabel 3.3 Keterangan Kode Kasus Penggunaan

Kode Kasus Penggunaan	Kasus Penggunaan	Aktor
UC-001	<i>Mengontrol jemuran pakaian secara otomatis</i>	Pengguna
UC-002	<i>Memantau kondisi cuaca</i>	Pengguna
UC-003	<i>Mengontrol posisi jemuran pakaian melalui smartphone</i>	Pengguna

3.2.2 Perancangan Aplikasi

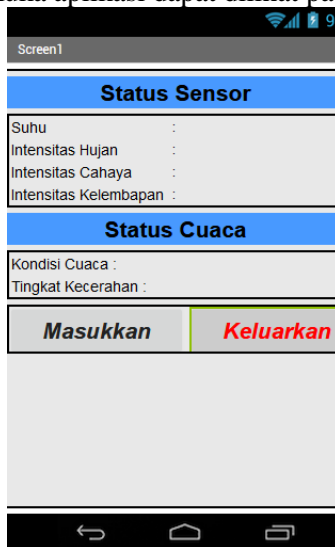
Pada tahap ini akan dijelaskan perancangan aplikasi untuk penunjang jemuran pakaian otomatis berbasis arduino ini. Dalam membangun aplikasi ini akan digunakan *MIT App Inventor*. Adapun layanan *cloud* yang digunakan untuk mengirim dan memvisualisasikan data adalah *Thinkspeak*. *ThingSpeak*™ memungkinkan pengguna untuk mengumpulkan, memvisualisasikan, dan menganalisis aliran data. *ThingSpeak* memberikan visualisasi instan terhadap data yang diposting oleh perangkat pengguna.



Gambar 3.9 Tampilan Interface ThingSpeak™ [12]

3.2.3 Perancangan Antarmuka Aplikasi

Berikut adalah perancangan antarmuka aplikasi jemuran pakaian otomatis ini. Dari rancangan ini pengguna dapat melakukan pemantauan kondisi cuaca terkait dengan intensitas suhu, hujan, intensitas cahaya, dan intensitas kelembapan. Rancangan antarmuka aplikasi dapat dilihat pada **Gambar 3.10**



Gambar 3.10 Rancangan antarmuka aplikasi

[Halaman ini sengaja dikosongkan]

BAB IV

IMPLEMENTASI SISTEM

Bab ini membahas implementasi dari perancangan sistem jemuran pakaian otomatis menggunakan motor dan sensor berbasis arduino yang di dalamnya akan dibahas mengenai lingkungan pengembangan sistem, implementasi sistem dan antarmuka pengguna.

4.1 Lingkungan Pengembangan Sistem

Lingkungan pengembangan sistem yang digunakan untuk mengembangkan Tugas Akhir ini dilakukan pada lingkungan dan kakas sebagai berikut.

Tabel 4.1 Lingkungan Pengembangan Sistem

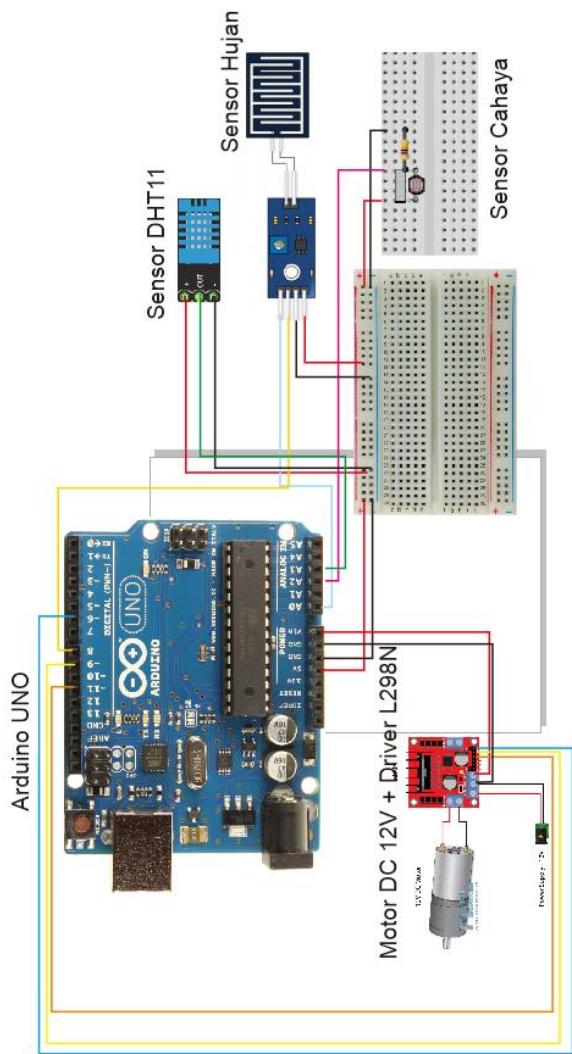
Perangkat	Jenis Perangkat	Spesifikasi
Perangkat keras	Prosesor	Intel® Core™ i5-3337U (1.80GHz)
	RAM	8 GB
	System Type	64-bit <i>Operating System</i>
Perangkat lunak	Sistem Operasi	Windows 10 Enterprise N
	IDE	Arduino <i>Software IDE</i> , <i>MIT App Inventor</i>

4. 2 Implementasi Alat

Pada bagian ini akan dijelaskan secara detail mengenai implementasi alat pada sistem jemuran pakaian otomatis berbasis arduino.

4.2.1 Gambar Rangkaian alat

Pada tahap implementasi ini akan dijabarkan bagaimana proses-proses merangkai alat jemuran pakaian otomatis berbasis arduino ini. Gambar rangkaian alat dapat dilihat pada **Gambar 4.1**.



Gambar 4.1 Rangkaian Alat

4. 3 Implementasi *Fuzzy Logic*

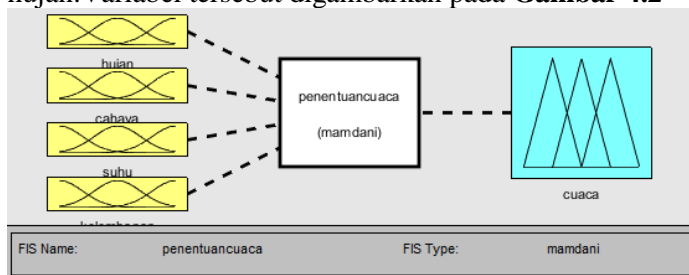
Pada subbbab Implementasi *Fuzzy Logic* akan menjelaskan tentang implementasi *fuzzy logic* dalam pengambilan keputusan. Keputusan ini digunakan untuk menentukan cuaca yang sedang terjadi sehingga sistem dapat menentukan kapan jemuran digerakkan ke tempat teduh dan kapan jemuran berada pada *normal state*. Penentuan kondisi cuaca menggunakan beberapa variabel yang mempengaruhi proses terjadinya hujan. Langkah-langkah dalam memprediksi curah hujan dilakukan dengan menerapkan *Fuzzy Inference System* tipe Mamdani, yaitu:

1. Membandingkan variabel-variabel *input* dengan *membership function* pada *anteseden*

Membandingkan variabel-variabel *input* dengan *membership function* pada *anteseden* dinamakan proses *fuzzyfikasi*. Tahapan untuk mendapatkan nilai keanggotaan masing-masing variabel linguistik sebagai berikut:

- a) Membentuk variabel *input* dan variabel *output*

Variabel input berupa hujan, cahaya, suhu dan kelembapan, sedangkan variabel *output* berupa cuaca yang terdiri dari tidak hujan, peringatan hujan dan hujan. Variabel tersebut digambarkan pada **Gambar 4.2**



Gambar 4.2 Variabel Input-Output Fuzzy

b) Membentuk himpunan *fuzzy*

Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*. Variabel linguistik yang digunakan sebagai berikut:

a. Variabel *Input*

Semua variabel input membentuk himpunan *fuzzy* dan semesta pembicaraan dijelaskan pada **Tabel 4.2**

Tabel 4.2 Variabel *Input*

Variabel <i>Input</i> Hujan	{Hujan, Tidak Hujan }
Variabel <i>Input</i> Cahaya	{Gelap, Terang }
Variabel <i>Input</i> Suhu	{Dingin, Normal }
Variabel <i>Input</i> Kelembapan	{Lembap, Kering }

b. Variabel *Output*

Semua variabel output membentuk himpunan *fuzzy* dan semesta pembicaraan dijelaskan pada **Tabel**

Tabel 4.3 Variabel *Output*

Variabel <i>Output</i> Cuaca	{Hujan, Peringatan Hujan, Tidak Hujan }
------------------------------	---

c) Membentuk himpunan semesta pembicaraan masing-masing variabel

Semesta pembicaraan tiap-tiap variabel merupakan rentang kemungkinan tiap hari. Semesta pembicaraan ditunjukkan pada **Tabel 4.4**

Tabel 4.4 Himpunan Pembicaraan Variabel

Fungsi	Variabel	Semesta Pembicaraan
Input	Hujan	[0 1024]
	Cahaya	[0 30]
	Suhu	[0 40]
	Kelembapan	[0 30]
Output	Cuaca	[0 10]

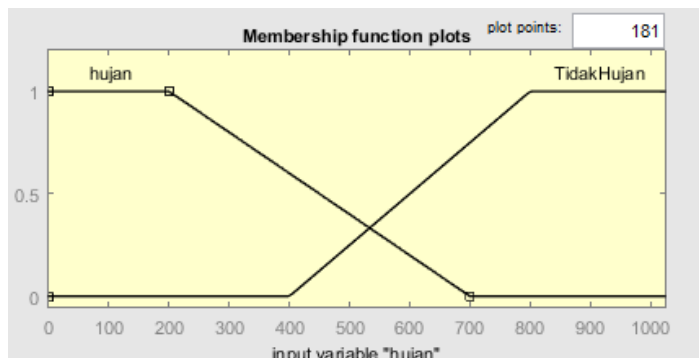
d) Menentukan fungsi keanggotaan tiap-tiap variabel

Jika ingin mendapatkan nilai keanggotaan, maka dapat menentukan fungsi keanggotaan tiap-tiap variabel dengan melalui pendekatan fungsi. Fungsi keanggotaan yang digunakan dalam penelitian ini adalah *Trapezium Member Function*.

a. Variabel Hujan

Variabel Hujan membentuk himpunan *fuzzy* dengan fungsi keanggotannya sebagai berikut:

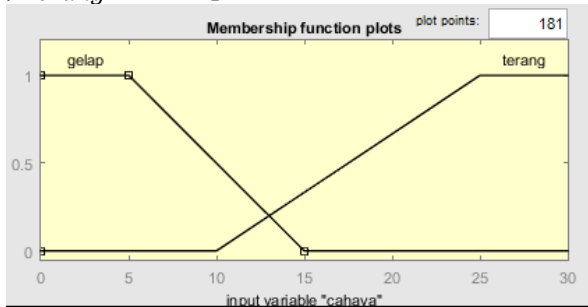
- $\mu_{Hujan}(x) = \text{trapmf}(x; 0, 0, 200, 700)$
- $\mu_{TidakHujan}(x) = \text{trapmf}(x; 400, 800, 1024, 1024)$

**Gambar 4.3 Himpunan Keanggotaan Variabel Hujan**

b. Variabel Cahaya

Variabel Cahaya membentuk himpunan *fuzzy* dengan fungsi keanggotannya sebagai berikut:

- $\mu_{Gelap}(x) = \text{trapmf}(x; 0, 0, 5, 15)$
- $\mu_{Terang}(x) = \text{trapmf}(x; 10, 25, 30, 50)$

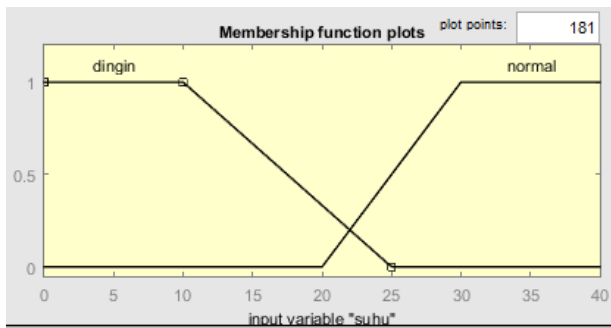


Gambar 4.4 Himpunan Keanggotaan Variabel Cahaya

c. Variabel Suhu

Variabel Suhu membentuk himpunan *fuzzy* dengan fungsi keanggotannya sebagai berikut:

- $\mu_{Dingin}(x) = \text{trapmf}(x; -10, 0, 10, 25)$
- $\mu_{Normal}(x) = \text{trapmf}(x; 20, 30, 40, 60)$

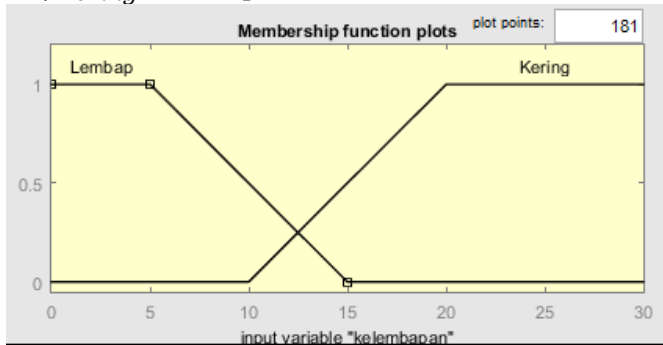


Gambar 4.5 Himpunan Keanggotaan Variabel Suhu

d. Variabel Kelembapan

Variabel Kelembapan membentuk himpunan *fuzzy* dengan fungsi keanggotannya sebagai berikut:

- $\mu_{Lembap}(x) = \text{trapmf}(x; -10, 0, 5, 15)$
- $\mu_{Kering}(x) = \text{trapmf}(x; 10, 20, 30, 40)$

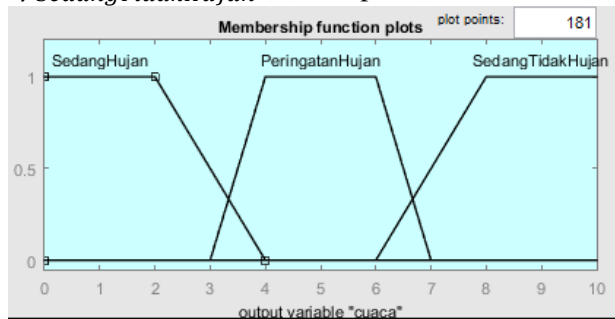


Gambar 4.6 Himpunan Keanggotaan Variabel Kelembapan

e. Variabel Cuaca

Variabel Kelembapan membentuk himpunan *fuzzy* dengan fungsi keanggotannya sebagai berikut:

- $\mu_{SedangHujan}(x) = \text{trapmf}(x; 0, 0, 2, 4)$
- $\mu_{PeringatanHujan}(x) = \text{trapmf}(x; 3, 4, 6, 7)$
- $\mu_{SedangTidakHujan}(x) = \text{trapmf}(x; 6, 8, 10, 10)$



Gambar 4.7 Himpunan Keanggotaan Variabel Cuaca

2. Mengkombinasikan semua variabel *input* dengan menerapkan *t-norm*

Langkah selanjutnya adalah mengkombinasi semua variabel input dengan menerapkan *t-norm*. Langkah ini biasanya disebut pembentukan fungsi implikasi. Pada penelitian ini digunakan operator min atau multikonjungtif dengan penghubung “AND”.

3. Menghasilkan konsekuensi yang memenuhi syarat atau masing-masing aturan berdasar bobotnya

Setelah menentukan fungsi implikasi, langkah selanjutnya yaitu pembentukan basis aturan *fuzzy*. Basis aturan fuzzy merupakan keseluruhan aturan dari kombinasi enam input yang mungkin dengan masing-masing tiga fungsi keanggotaan dan menghasilkan konsekuensi yang bobotnya sesuai.

Pembentukan *rules* merupakan langkah penting dalam metode FIS tipe Mamdani karena ketepatan defuzzyfikasi. Bentuk If-Then merupakan bentuk pernyataan sistem berbasis penalaran.

Setiap aturan dapat dituliskan dalam bentuk bahasa berikut:
“If premis, then konsekuensi.”

atau dapat didefinisikan juga sebagai berikut:

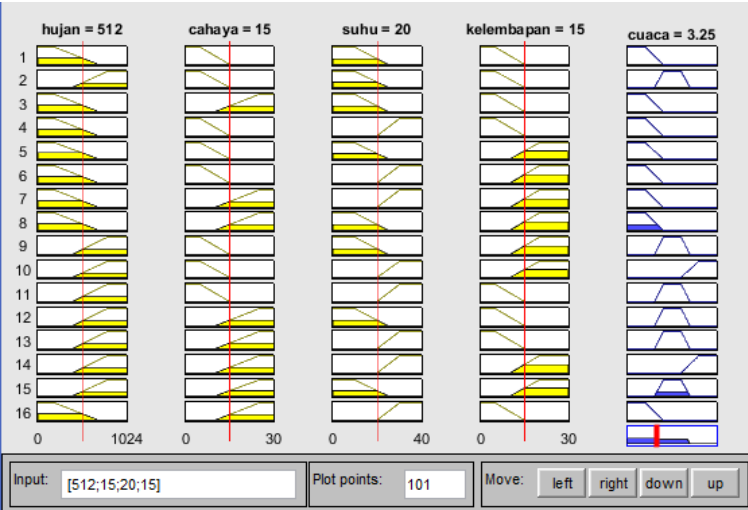
IF x_1 is A_1 AND...AND x_n is A_n THEN y is B

Dalam pengerjaan Sistem Jemuran Pakaian Otomatis Menggunakan Sensor dan Motor Berbasis Arduino dihasilkan beberapa *rules* yang dijelaskan pada **Tabel 4.5**

Tabel 4.5 Rules

<i>Rule 1</i>	If (hujan is hujan) and (cahaya is gelap) and (suhu is dingin) and (kelembapan is Lembab) then (cuaca is SedangHujan) (1)
<i>Rule 2</i>	If (hujan is hujan) and (cahaya is gelap) and (suhu is dingin) and (kelembapan is Kering) then (cuaca is SedangHujan) (1)
<i>Rule 3</i>	If (hujan is hujan) and (cahaya is gelap) and (suhu is normal) and (kelembapan is Lembab) then (cuaca is SedangHujan) (1)
<i>Rule 4</i>	If (hujan is hujan) and (cahaya is gelap) and (suhu is normal) and (kelembapan is Kering) then (cuaca is SedangHujan) (1)
<i>Rule 5</i>	If (hujan is hujan) and (cahaya is terang) and (suhu is normal) and (kelembapan is Kering) then (cuaca is SedangHujan) (1)
<i>Rule 6</i>	If (hujan is hujan) and (cahaya is terang) and (suhu is normal) and (kelembapan is Lembab) then (cuaca is SedangHujan) (1)
<i>Rule 7</i>	If (hujan is hujan) and (cahaya is terang) and (suhu is dingin) and (kelembapan is Lembab) then (cuaca is SedangHujan) (1)
<i>Rule 8</i>	If (hujan is hujan) and (cahaya is terang) and (suhu is dingin) and (kelembapan is Kering) then (cuaca is SedangHujan) (1)
<i>Rule 9</i>	If (hujan is TidakHujan) and (cahaya is gelap) and (suhu is dingin) and (kelembapan is Lembab) then (cuaca is PeringatanHujan) (1)
<i>Rule 10</i>	If (hujan is TidakHujan) and (cahaya is gelap) and (suhu is dingin) and (kelembapan is Kering) then (cuaca is PeringatanHujan) (1)
<i>Rule 11</i>	If (hujan is TidakHujan) and (cahaya is gelap) and (suhu is normal) and (kelembapan is Kering) then (cuaca is SedangTidakHujan) (1)
<i>Rule 12</i>	If (hujan is TidakHujan) and (cahaya is gelap) and (suhu is normal) and (kelembapan is Lembab) then (cuaca is PeringatanHujan) (1)

<i>Rule 13</i>	If (hujan is TidakHujan) and (cahaya is terang) and (suhu is normal) and (kelembapan is Lembab) then (cuaca is PeringatanHujan) (1)
<i>Rule 14</i>	If (hujan is TidakHujan) and (cahaya is terang) and (suhu is normal) and (kelembapan is Kering) then (cuaca is SedangTidakHujan) (1)
<i>Rule 15</i>	If (hujan is TidakHujan) and (cahaya is terang) and (suhu is dingin) and (kelembapan is Kering) then (cuaca is PeringatanHujan) (1)
<i>Rule 16</i>	If (hujan is TidakHujan) and (cahaya is terang) and (suhu is dingin) and (kelembapan is Lembab) then (cuaca is PeringatanHujan) (1)



Gambar 4.8 Tampilan *Fuzzy Rule Viewer*

4. Agregasi seluruh bagian konsekuensi

Langkah selanjutnya dalam penentuan kondisi cuaca dengan menerapkan metode *Fuzzy Inference System* tipe Mamdani adalah defuzzyfikasi. Defuzzyfikasi merupakan proses mengubah himpunan *fuzzy* keluaran menjadi keluaran tegas (*crisp*). Nilai tegas keluaran diperoleh dari himpunan-himpunan *fuzzy* keluaran dengan menggunakan metode *Centroid*.

5. Implementasi *Coding Fuzzy* pada Arduino

Langkah terakhir dalam implementasi *fuzzy logic* adalah dengan mengimplementasikannya pada *microcontroller* Arduino. Untuk melakukan implementasi *coding* pada tugas akhir ini digunakan *eFLL (Embedded Fuzzy Logic Library)*. Dalam implementasi *fuzzy logic*, langkah awal yang dilakukan adalah melakukan inisiasi himpunan *input fuzzy*. Berikut adalah kode sumber untuk melakukan inisiasi himpunan *input fuzzy*.

```
// Inisiasi Fuzzy
Fuzzy* fuzzy = new Fuzzy();

//Himpunan Variabel Hujan
FuzzySet* Hujan = new FuzzySet(0, 0, 200, 700);
FuzzySet* TidakHujan = new FuzzySet(400, 800, 1024, 1024);

//Himpunan Variabel Cahaya
FuzzySet* Gelap = new FuzzySet(0, 0, 5, 15);
FuzzySet* Terang = new FuzzySet(10, 25, 30, 50);

//Himpunan Variabel Suhu
FuzzySet* Dingin = new FuzzySet(-10, 0, 10, 25);
FuzzySet* Normal = new FuzzySet(20, 30, 40, 60);

//Himpunan Variabel Kelembapan
FuzzySet* Lembap = new FuzzySet(-10, 0, 5, 15);
```

```
FuzzySet* Kering = new FuzzySet(10, 20, 30, 40);
```

Kode Sumber 4.1 Inisiasi Input Fuzzy

Langkah selanjutnya yang dilakukan adalah melakukan inisiasi himpunan *output fuzzy*. Berikut adalah kode sumber untuk melakukan inisiasi himpunan *output fuzzy*.

```
// FuzzyOutput Cuaca
FuzzyOutput* cuaca = new FuzzyOutput(1);
FuzzySet* SedangHujan = new FuzzySet(0, 0, 2, 4);
cuaca->addFuzzySet(SedangHujan);
FuzzySet* PeringatanHujan = new FuzzySet(3, 4, 6, 7);
cuaca->addFuzzySet(PeringatanHujan);
FuzzySet* SedangTidakHujan = new FuzzySet(6, 8, 10, 10);
cuaca->addFuzzySet(SedangTidakHujan);
fuzzy->addFuzzyOutput(cuaca);
```

Kode Sumber 4.2 Inisiasi Output Fuzzy

Setelah melakukan inisiasi langkah selanjutnya yang dilakukan adalah mendefinisikan aturan-aturan (*rules*) untuk menentukan langkah pengambilan keputusan. Pada Tugas Akhir kali ini digunakan untuk menentukan kondisi cuaca yang sedang terjadi. Berikut adalah contoh kode sumber untuk mendefinisikan aturan-aturan (*rules*) :

```
// Fuzzy Rule 1 ( HujanDeras && Gelap && Dingin && Lembap )
FuzzyRuleAntecedent* hujanHujanDerasAndcahayaGelap = new
FuzzyRuleAntecedent();
hujanHujanDerasAndcahayaGelap->joinWithAND(HujanDeras,
Gelap);

FuzzyRuleAntecedent* suhuDinginAndkelembapanLembap =
new FuzzyRuleAntecedent();
suhuDinginAndkelembapanLembap->joinWithAND(Dingin,
Lembap);
```



```

FuzzyRuleAntecedent*
ifhujanHujanDerasAndcahayaGelapAndsuhuDinginAndkelembap
anLembap = new FuzzyRuleAntecedent();

ifhujanHujanDerasAndcahayaGelapAndsuhuDinginAndkelembap
anLembap->joinWithAND(hujanHujanDerasAndcahayaGelap,
suhuDinginAndkelembapanLembap);

FuzzyRuleConsequent*   thencuacaSedangHujan   =   new
FuzzyRuleConsequent();
thencuacaSedangHujan->addOutput(SedangHujan);

FuzzyRule*   fuzzyRule1   =   new   FuzzyRule(1,
ifhujanHujanDerasAndcahayaGelapAndsuhuDinginAndkelembap
anLembap, thencuacaSedangHujan);
fuzzy->addFuzzyRule(fuzzyRule1);

```

Kode Sumber 4.3 Definisi Rule Fuzzy

4. 3 Implementasi *Coding* Alat

Pada subbbab Implementasi *Coding* Alat akan menjelaskan tentang pembuatan perangkat yang telah direncanakan pada Bab III. Berikut ini adalah implementasi *coding* terhadap alat. Dalam proses *coding* dikerjakan menggunakan Arduino IDE. Detail pembahasan mengenai kode sumber yang dibuat dalam sistem akan dijelaskan pada sub-bab berikut.

4.3.1 Kode Sumber Pembacaan Sensor

Bagian ini adalah implementasi dari pembacaan sensor pada Sistem Jemuran Pakaian Otomatis. Dalam hal ini ada beberapa sensor yang digunakan yaitu sensor hujan (*Raindrop Sensor*), sensor kelembapan (DHT11), dan sensor LDR (*Light Dependent Resistor*). Nantinya data yang dibaca oleh sensor akan ditampilkan pada *Serial Monitor* yang ada pada Arduino IDE. Berikut adalah kode sumber dari pembacaan sensor tertera pada **Kode Sumber 4.4**

```
void loop() {
```

```
sensor = analogRead(OutputA0);
ldrSensor = analogRead(OutputLDR);

float h = dht.readHumidity();
float t = dht.readTemperature();
float f = dht.readTemperature(true);

if (isnan(h) || isnan(t) || isnan(f)) {
  Serial.println("Failed to read from DHT sensor!");
  return;
}

float hic = dht.computeHeatIndex(t, h, false);
Serial.print(" Hujan: ");
Serial.print(sensor);
Serial.print(" Ldr: ");
Serial.print(ldrSensor);
Serial.print(" Temp:");
Serial.print(t);
Serial.print(" Humidity:");
Serial.print(h);
Serial.print("\n");
}
```

Kode Sumber 4.4 Pembacaan Sensor

4.3.2 Kode Sumber Pengiriman Data Sensor

Bagian ini adalah implementasi dari pengiriman data sensor pada Sistem Jemuran Pakaian Otomatis. Dalam hal ini ada beberapa sensor yang digunakan yaitu sensor hujan (*Raindrop Sensor*), sensor kelembapan (DHT11), dan sensor LDR (*Light Dependent Resistor*). Nantinya data yang dibaca oleh sensor akan dikirim menuju server melalui kode berikut. Berikut adalah kode sumber dari pengiriman data sensor :

```
void loop(){
String cmd = "AT+CIPSTART=\\"TCP\\","\\"";
cmd += "184.106.153.149"; // api.thingspeak.com
cmd += "\\"",80";
ser.println(cmd);

if(ser.find("Error")){
  Serial.println("AT+CIPSTART error");
  return;
}

//kirim data ke thinkspeak
String getStr = "GET /update?api_key=";
getStr += apiKey;
getStr += "&field1=";
getStr += String(t);
getStr += "&field4=";
getStr += String(h);
getStr += "\r\n\r\n";

cmd = "AT+CIPSEND=";
cmd += String(getStr.length());
ser.println(cmd);

if(ser.find(">")){
  ser.print(getStr);
}
else{
```

```

    ser.println("AT+CIPCLOSE");
    Serial.println("AT+CIPCLOSE");
  }
  delay(16000);
}

```

Kode Sumber 4.5 Pengiriman Data Sensor

4.3.2 Kode Sumber Logika Jemuran Otomatis

Bagian ini adalah implementasi dari logika pada Sistem Jemuran Pakaian Otomatis. Apabila sensor mendeteksi kondisi gelap ataupun hujan maka secara otomatis menggerakkan jemuran ke tempat yang teduh. Sebaliknya, bila kondisi terang maka jemuran bergerak ke *norma state*. Berikut adalah kode sumber dari logika sistem jemuran pakaian otomatis :

```

if((valLDR < 10) and (mark == 0)){
  delay(200);
  mark = 1;
  Serial.println(" == MEMASUKKAN ==");
  teduh();
}

if((valHujan == LOW) and (mark == 0)){
  delay(200);
  mark = 2;
  Serial.println(" == MEMASUKKAN ==");
  teduh();
}

if((valLDR > 10) and (valHujan == HIGH) and (mark > 0)){
  delay(200);
  mark = 0;
  Serial.println(" == MENGELUARKAN ==");
  normal();
}

```

Kode Sumber 4.6 Logika Jemuran Otomatis

4.3.3 Kode Sumber Menggerakkan Jemuran ke Tempat Teduh

Bagian ini adalah implementasi dari menggerakkan jemuran ke tempat teduh pada Sistem Jemuran Pakaian Otomatis. Berikut adalah kode sumber menggerakkan jemuran ke tempat teduh :

```
void teduh() {
  analogWrite(ENB, SPEED);
  digitalWrite(In3, LOW);
  digitalWrite(In4, HIGH);

  int button2 = digitalRead(InputB2);
  if(button2 == HIGH) {
    analogWrite(ENB, 0);
    digitalWrite(In3, LOW);
    digitalWrite(In4, LOW);
    delay(1000);
    return;
  }

  // teduh();
}
```

Kode Sumber 4.7 Menggerakkan Jemuran ke Tempat Teduh

4.3.4 Kode Sumber Menggerakkan Jemuran ke *Normal State*

Bagian ini adalah implementasi dari menggerakkan jemuran ke *normal state* pada Sistem Jemuran Pakaian Otomatis. Berikut adalah kode sumber menggerakkan jemuran ke tempat *normal state* tertera pada **Kode Sumber 4.8**

```

void normal() {
    analogWrite(ENB, SPEED);
    digitalWrite(In3, HIGH);
    digitalWrite(In4, LOW);

    int button1 = digitalRead(InputB1);
    if(button1 == HIGH) {
        analogWrite(ENB, 0);
        digitalWrite(In3, LOW);
        digitalWrite(In4, LOW);
        delay(1000);
        return;
    }
}

```

Kode Sumber 4.8 Menggerakkan Jemuran ke *normal state*

4.3.5 Kode Sumber Pengiriman Notifikasi Status Jemuran Pakaian Melalui *E-mail*

Bagian ini adalah implementasi dari pengiriman notifikasi status jemuran melalui *e---mail* pada Sistem Jemuran Pakaian Otomatis. Notifikasi ini diperlukan agar pengguna mengetahui kondisi jemuran pakaiannya sedang berada. Berikut akan dijelaskan beberapa kode sumber yang digunakan.

Untuk bisa mengirim sebuah pesan notifikasi *e-mail* maka perangkat harus dihubungkan menuju jaringan internet. Kali ini *modul wifi* digunakan untuk menyambungkan perangkat ke jaringan internet. Agar terhubung maka diperlukan sebuah *access point* yang didefinisikan pada **Kode Sumber 4.9**

```

#pragma region Globals
const char* ssid = "iot";           // WIFI network name
const char* password = "abnabnabn"; // WIFI network password
uint8_t connection_state = 0;       // Connected to WIFI or not
uint16_t reconnect_interval = 10000; // If not connected wait time
#pragma endregion Globals

```

Kode Sumber 4.9 Menyambungkan Modul Wifi menuju Access Point

Untuk bisa mengirim sebuah pesan notifikasi *e-mail* maka diperlukan alamat email pengirim yang didefinisikan pada **Kode Sumber 4.10**

```
class Gsender
{
    protected:
        Gsender();
    private:
        const int SMTP_PORT = 465;
        const char* SMTP_SERVER = "smtp.gmail.com";
        const char* EMAILBASE64_LOGIN =
            "amVtdXJhbm90b2lhdGlzQGdtYWlsLmNvbQ==";
        const char* EMAILBASE64_PASSWORD =
            "Mm5vdmVtYmVyOTY=";
        const char* FROM = "jemuranotomatis@gmail.com";
        const char* _error = nullptr; char* _subject = nullptr;
        String _serverResponse; static Gsender* _instance;
        bool AwaitSMTPResponse(WiFiClientSecure &client, const
            String &resp = "", uint16_t timeOut = 10000);
    public:
        static Gsender* Instance();
        Gsender* Subject(const char* subject);
        Gsender* Subject(const String &subject);
        bool Send(const String &to, const String &message);
        String getLastResponse(); const char* getError();
};
```

Kode Sumber 4.10 Definisi E-mail Pengirim Notifikasi

Berikut adalah kode sumber untuk mengirim sebuah pesan notifikasi *e-mail* ketika jemuran dikeluarkan

```
Gsender *gsender = Gsender::Instance(); // Getting pointer to class
String subject = "Informasi Status Jemuran";
gsender->Subject(subject)-
>Send("bagus.nusantara313@gmail.com", "Jemuran dikeluarkan
sistem menuju ke normal state");
```

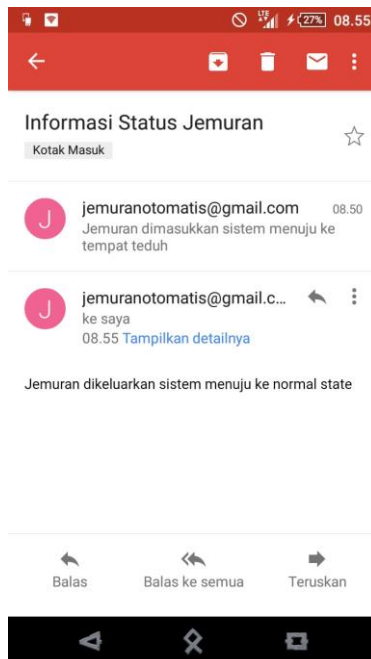
Kode Sumber 4.11 Notifikasi Status Jemuran ketika Dikeluarkan

Berikut adalah kode sumber untuk mengirim sebuah pesan notifikasi *e-mail* ketika jemuran dimasukkan.

```
Gsender *gsender = Gsender::Instance(); // Getting pointer to class instance
String subject = "Informasi Status Jemuran";
gsender->Subject(subject)-
>Send("bagus.nusantara313@gmail.com", "Jemuran dimasukkan sistem menuju ke tempat teduh")
```

Kode Sumber 4.12 Notifikasi Status Jemuran ketika Dimasukkan

Berikut adalah hasil implementasi notifikasi *e-mail* yang dikirimkan sistem menuju *e-mail* pengguna.



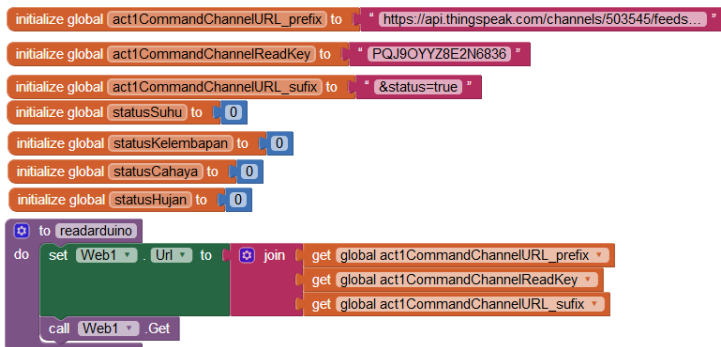
Gambar 4.9 Hasil Implementasi Notifikasi *E-mail*

4.4 Implementasi Pembuatan Aplikasi *Mobile*

Pada bagian ini akan dijelaskan mengenai implementasi pembuatan Aplikasi *Mobile* jemuran pakaian otomatis berbasis arduino. Aplikasi ini dirancang menggunakan *MIT App Inventor 2*. Kode sumber implementasi pembuatan Aplikasi *Mobile* dapat dilihat pada gambar berikut :

4.4.1 Kode Sumber Menyambungkan API *Thinkspeak* menuju Aplikasi *Mobile*

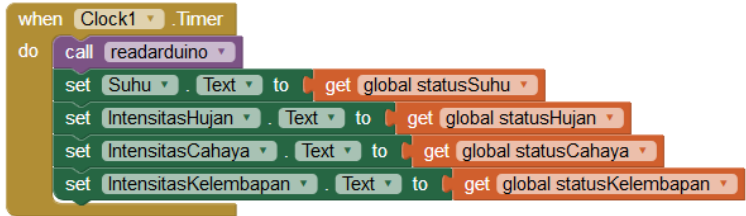
Bagian ini adalah implementasi koding untuk menyambungkan *API Thinkspeak* menuju aplikasi *mobile*. Rangkaian kode dapat dilihat pada **Gambar**



Gambar 4.10 Kode Sumber Menyambungkan API Thinkspeak

4.4.2 Kode Sumber Menampilkan Nilai Sensor pada Aplikasi *Mobile*

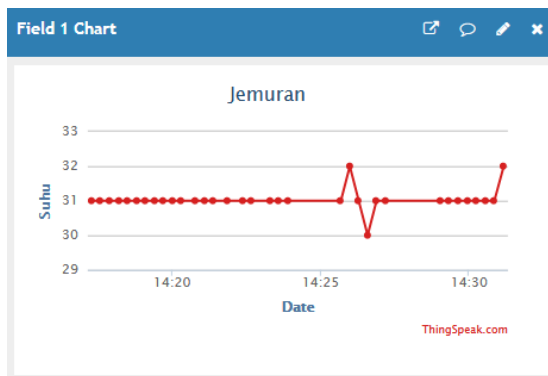
Bagian ini adalah implementasi koding untuk menampilkan nilai data sensor aplikasi *mobile*. Rangkaian kode dapat dilihat pada **Gambar**



Gambar 4.11 Kode Sumber Menampilkan Nilai Sensor

4.4.3 Hasil Pembacaan Data Sensor Suhu pada Aplikasi *Thinkspeak*

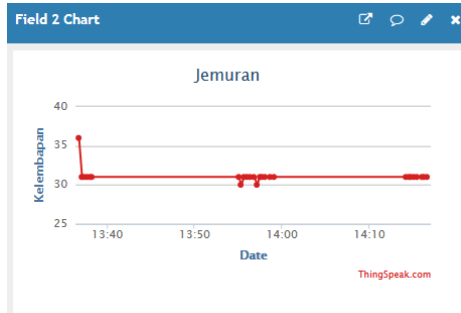
Bagian ini menampilkan grafik hasil pembacaan sensor suhu pada aplikasi *Thinkspeak*.



Gambar 4.12 Hasil Pembacaan Data Sensor Suhu

4.4.4 Hasil Pembacaan Data Sensor Kelembapan pada Aplikasi *Thinkspeak*

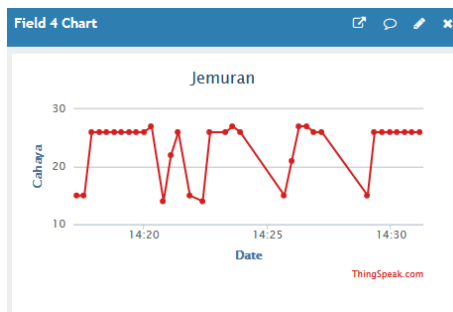
Bagian ini menampilkan grafik hasil pembacaan sensor kelembapan pada aplikasi *Thinkspeak*.



Gambar 4.13 Hasil Pembacaan Data Sensor Kelembapan

4.4.5 Hasil Pembacaan Data Sensor Cahaya pada Aplikasi *Thinkspeak*

Bagian ini menampilkan grafik hasil pembacaan sensor cahaya pada aplikasi *Thinkspeak*.



Gambar 4.14 Hasil Pembacaan Data pada Sensor Cahaya

4.4.6 Implementasi Tampilan Aplikasi *Mobile*

Bagian ini menampilkan hasil implementasi tampilan aplikasi *mobile* Sistem Jemuran Pakaian Otomatis Berbasis Arduino.



Gambar 4.15 Implementasi Tampilan Aplikasi Mobile

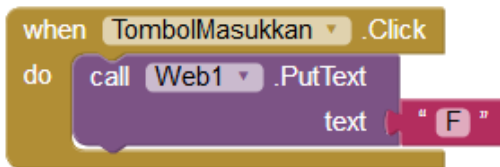
4.4.7 Kode Sumber Menggerakkan Jemuran ke Tempat Teduh Menggunakan Aplikasi *Mobile*

Bagian ini adalah implementasi dari menggerakkan jemuran ke tempat teduh pada Sistem Jemuran Pakaian Otomatis menggunakan aplikasi *Mobile*. Berikut adalah kode sumber menggerakkan jemuran ke tempat teduh :

```
void loop() {
  if (Serial.available() > 0){
    state = Serial.read();
    flag=0;
  }
  if (state == 'F') {
    delay(200);
    mark = 1;
    Serial.println(" == MEMASUKKAN ==");
    teduh();
  }
}
```

Kode Sumber 4.13 Menggerakkan Jemuran ke Tempat Teduh Menggunakan Aplikasi Mobile

Bagian ini adalah implementasi dari menggerakkan jemuran ke tempat teduh pada Sistem Jemuran Pakaian Otomatis menggunakan aplikasi *Mobile* menggunakan Aplikasi *MIT App Inventor*. Berikut adalah kode sumber menggerakkan jemuran ke tempat teduh :



Gambar 4.16 Kode Sumber Memasukkan Jemuran

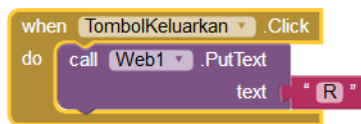
4.4.7 Kode Sumber Menggerakkan Jemuran ke Normal State Menggunakan Aplikasi Mobile

Bagian ini adalah implementasi dari menggerakkan jemuran ke *normal state* pada Sistem Jemuran Pakaian Otomatis menggunakan aplikasi *Mobile*. Berikut adalah kode sumber menggerakkan jemuran ke *normal state* :

```
void loop() {
  if (Serial.available() > 0){
    state = Serial.read();
    flag=0;
  }
  if (state == 'R') {
    delay(200);
    mark = 0;
    Serial.println(" == MENGELUARKAN ==");
    normal();
  }
}
```

Kode Sumber 4.14 Menggerakkan Jemuran ke Tempat Teduh Menggunakan Aplikasi Mobile

Bagian ini adalah implementasi dari menggerakkan jemuran ke *normal state* pada Sistem Jemuran Pakaian Otomatis menggunakan aplikasi *Mobile* menggunakan Aplikasi *MIT App Inventor*. Berikut adalah kode sumber menggerakkan jemuran ke *normal state*:



Gambar 4.17 Kode Sumber Mengeluarkan Jemuran

4.4 Implementasi Pembuatan *Prototype*

Pada bagian ini akan dijelaskan mengenai hasil implementasi pembuatan *prototype* jemuran pakaian otomatis

berbasis arduino. *Prototype* ini dibuat menggunakan kayu dan kertas *sticker*. Untuk menggerakkan jemuran digunakan *GT2 pulley* beserta *timing belt*nya.



Gambar 4.18 Implementasi Pembuatan *Prototype*

[Halaman ini sengaja dikosongkan]

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada sistem yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan analisa sistem yang telah dijabarkan pada Bab III dan terhadap tujuan dibuatnya sistem jemuran pakaian otomatis berbasis arduino ini.

5.1 Skenario Pengujian

Pada bagian ini akan dijelaskan tentang skenario pengujian yang dilakukan. Pengujian dibagi menjadi dua yaitu pengujian alat dan pengujian perangkat lunak. Pengujian alat dilakukan dengan cara memberi rangsangan pada sensor dan melihat respon yang diberikan oleh sistem.

5.2 Pengujian Alat

Pengujian fungsionalitas alat dilakukan dengan melakukan skenario yang sama dengan rancangan alur proses aplikasi sebagai tolok ukur keberhasilan pengujian, dan mengacu pada kasus penggunaan yang sebelumnya telah dijelaskan pada Bab III. Pengujian pada alat dapat dijabarkan pada subbab berikut :

5.2.1 Pengujian *Fuzzy Logic* dalam Penentuan Kondisi Cuaca

Dalam membuat Sistem Jemuran Pakaian Otomatis ini digunakan *fuzzy logic* untuk menentukan kondisi cuaca. Rincian hasil pengujian *fuzzy logic* dalam penentuan kondisi cuaca pada sistem dapat dilihat pada **Tabel 5.1**.

Tabel 5.1 Hasil Pengujian *Fuzzy Logic*

Input				Output	
Hujan	Cahaya	Suhu	Kelembapan	Hasil	Label
300	10	1	5	Hujan: 0.80, 0.00 Cahaya: 0.50, 0.00 Suhu: 1.00, 0.00 Kelembapan: 1.00, 0.00 Kondisi Cuaca: 1.76	Sedang Hujan
900	11	15	10	Hujan: 0.00, 1.00 Cahaya: 0.40, 0.07 Suhu: 0.67, 0.00 Kelembapan: 0.50, 0.00 Kondisi Cuaca: 5.00	Pering atan Hujan
207	10	19	35	Hujan: 0.99, 0.00 Cahaya: 0.50, 0.00 Suhu: 0.40, 0.00 Kelembapan: 0.00, 0.50 Kondisi Cuaca: 1.81	Sedang Hujan
307	9	30	4	Hujan: 0.79, 0.00 Cahaya: 0.60, 0.00 Suhu: 0.00, 1.00 Kelembapan: 1.00, 0.00 Kondisi Cuaca: 1.72	Sedang Hujan
357	12	32	34	Hujan: 0.69, 0.00 Cahaya: 0.30, 0.13 Suhu: 0.00, 1.00 Kelembapan: 0.00, 0.60 Kondisi Cuaca: 1.85	Sedang Hujan
287	32	30	34	Hujan: 0.83, 0.00 Cahaya: 0.00, 0.90 Suhu: 0.00, 1.00 Kelembapan: 0.00, 0.60 Kondisi Cuaca: 1.72	Sedang Hujan
287	32	30	12	Hujan: 0.83, 0.00 Cahaya: 0.00, 0.90 Suhu: 0.00, 1.00 Kelembapan: 0.30, 0.20 Kondisi Cuaca: 1.90	Sedang Hujan
287	32	13	28	Hujan: 0.83, 0.00 Cahaya: 0.00, 0.90	Sedang Hujan

				Suhu: 0.80, 0.00 Kelembapan: 0.00, 1.00 Kondisi Cuaca: 1.63	
987	2	13	28	Hujan: 0.00, 1.00 Cahaya: 1.00, 0.00 Suhu: 0.80, 0.00 Kelembapan: 0.00, 1.00 Kondisi Cuaca: 5.00	Peringatan Hujan
987	2	33	28	Hujan: 0.00, 1.00 Cahaya: 1.00, 0.00 Suhu: 0.00, 1.00 Kelembapan: 0.00, 1.00 Kondisi Cuaca: 8.44	Sedang Tidak Hujan
1107	5	32	8	Hujan: 0.00, 1.00 Cahaya: 1.00, 0.00 Suhu: 0.00, 1.00 Kelembapan: 0.70, 0.00 Kondisi Cuaca: 5.00	Peringatan Hujan
1107	25	32	38	Hujan: 0.00, 1.00 Cahaya: 0.00, 1.00 Suhu: 0.00, 1.00 Kelembapan: 0.00, 0.20 Kondisi Cuaca: 8.10	Sedang Tidak Hujan

5.2.1 Pengujian Respon Alat terhadap Sensor Hujan (*Raindrop Sensor*)

Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada **Tabel 5.2**

Tabel 5.2 Pengujian Respon Alat terhadap Sensor Hujan

Nomor	UC-001
Nama	Pengujian Respon Alat terhadap Sensor Hujan
Tujuan Pengujian	Menguji respon sistem ketika sensor hujan ditetesi air
Skenario 1	Pengguna meneteskan air pada sensor hujan
Kondisi Awal	Jemuran berada pada <i>normal state</i>
Langkah Pengujian	Pengguna meneteskan air pada sensor hujan
Hasil Yang Diharapkan	Jemuran berpindah ke tempat yang teduh
Hasil Yang Didapat	Jemuran berpindah ke tempat yang teduh
Hasil Pengujian	Berhasil
Kondisi Akhir	Jemuran berpindah pada tempat yang teduh
Skenario 2	Pengguna membersihkan air pada sensor hujan
Kondisi Awal	Jemuran berada pada tempat yang teduh
Langkah Pengujian	Pengguna membersihkan air pada sensor hujan
Hasil Yang Diharapkan	Jemuran berpindah ke <i>normal state</i>
Hasil Yang Didapat	Jemuran berpindah ke <i>normal state</i>
Hasil Pengujian	Berhasil
Kondisi Akhir	Jemuran berpindah pada <i>normal state</i>

5.2.2 Pengujian Respon Alat terhadap Sensor Cahaya (*Light Dependent Resistor*)

Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada **Tabel 5.3**.

Tabel 5.3 Pengujian Respon Alat terhadap Sensor Cahaya

Nomor	UC-002
Nama	Pengujian Respon Alat terhadap Sensor Cahaya (<i>Light Dependent Resistor</i>)
Tujuan Pengujian	Menguji respon sistem ketika sensor cahaya mendeteksi keadaan gelap
Skenario 1	Pengguna menutup sensor cahaya sehingga sensor mendeteksi keadaan gelap
Kondisi Awal	Jemuran berada pada <i>normal state</i>
Langkah Pengujian	Pengguna menutup sensor cahaya
Hasil Yang Diharapkan	Jemuran berpindah ke tempat yang teduh
Hasil Yang Didapat	Jemuran berpindah ke tempat yang teduh
Hasil Pengujian	Berhasil
Kondisi Akhir	Jemuran berpindah pada tempat yang teduh
Skenario 2	Pengguna membiarkan sensor cahaya dalam keadaan terbuka sehingga sensor mendeteksi keadaan terang
Kondisi Awal	Jemuran berada pada tempat yang teduh
Langkah Pengujian	Pengguna membiarkan sensor cahaya dalam keadaan terbuka
Hasil Yang Diharapkan	Jemuran berpindah ke <i>normal state</i>
Hasil Yang Didapat	Jemuran berpindah ke <i>normal state</i>
Hasil Pengujian	Berhasil
Kondisi Akhir	Jemuran berpindah pada <i>normal state</i>

5.2.3 Pengujian Keberhasilan Aplikasi *Mobile* dalam Melakukan Pembacaan Data pada Sensor

Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada **Tabel 5.4**.

Tabel 5.4 Pengujian Keberhasilan Aplikasi Mobile dalam Melakukan Pembacaan Data pada Sensor

Nomor	UC-003
Nama	Pengujian Keberhasilan Alat dalam Melakukan Pembacaan Data pada Sensor
Tujuan Pengujian	Menguji keberhasilan alat dalam melakukan pembacaan data pada sensor
Skenario 1	Pengguna melihat nilai sensor pada aplikasi <i>mobile</i>
Kondisi Awal	Nilai data sensor pada aplikasi <i>mobile</i> menunjukkan nilai 0
Langkah Pengujian	Pengguna melihat aplikasi <i>mobile</i>
Hasil Yang Diharapkan	Aplikasi dapat menunjukkan nilai suhu, intensitas hujan, intensitas cahaya dan intensitas kelembapan
Hasil Yang Didapat	Aplikasi dapat menunjukkan nilai suhu, intensitas hujan, intensitas cahaya dan intensitas kelembapan
Hasil Pengujian	Berhasil
Kondisi Akhir	Aplikasi dapat menunjukkan nilai suhu, intensitas hujan, intensitas cahaya dan intensitas kelembapan

5.2.3 Pengujian Keberhasilan Sistem dalam Mengontrol Jemuran Pakaian Secara Manual Menggunakan Aplikasi Mobile

Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada **Tabel 5.5**.

Tabel 5.5 Pengujian Keberhasilan Sistem dalam Mengontrol Jemuran Pakaian Secara Manual Menggunakan Aplikasi Mobile

Nomor	UC-004
Nama	Pengujian Keberhasilan Sistem dalam Mengontrol Jemuran Pakaian Secara Manual Menggunakan Aplikasi <i>Mobile</i>
Tujuan Pengujian	Menguji Keberhasilan Sistem dalam Mengontrol Jemuran Pakaian Secara Manual Menggunakan Aplikasi Mobile
Skenario 1	Pengguna menekan tombol masukkan pada aplikasi <i>mobile</i>
Kondisi Awal	Jemuran berada pada <i>normal state</i>
Langkah Pengujian	Pengguna menekan tombol masukkan pada aplikasi <i>mobile</i>
Hasil Yang Diharapkan	Jemuran berpindah ke tempat yang teduh
Hasil Yang Didapat	Jemuran berpindah ke tempat yang teduh
Hasil Pengujian	Berhasil
Kondisi Akhir	Jemuran berpindah pada tempat yang teduh
Skenario 2	Pengguna menekan tombol Keluarkan pada aplikasi <i>mobile</i>
Kondisi Awal	Jemuran berada pada tempat yang teduh
Langkah Pengujian	Pengguna menekan tombol masukkan pada aplikasi <i>mobile</i>
Hasil Yang Diharapkan	Jemuran berpindah ke <i>normal state</i>
Hasil Yang Didapat	Jemuran berpindah ke <i>normal state</i>
Hasil Pengujian	Berhasil
Kondisi Akhir	Jemuran berpindah pada <i>normal state</i>

5.3 Evaluasi Pengujian Fungsionalitas

Rangkuman mengenai hasil pengujian fungsionalitas dapat dilihat pada **Tabel 5.6**. Berdasarkan data pada tabel tersebut, semua skenario pengujian berhasil dan program berjalan dengan baik. Sehingga bisa ditarik kesimpulan bahwa fungsionalitas dari aplikasi telah dapat bekerja sesuai dengan yang diharapkan.

Tabel 5.6 Evaluasi Pengujian Fungsionalitas

No.	Kode Kasus Penggunaan	Nama Kasus Penggunaan	Skenario	Hasil
1	UC-001	<i>Pengujian Respon Alat terhadap Sensor Hujan</i>	Skenario 1	Berhasil
			Skenario 2	Berhasil
2	UC-002	<i>Pengujian Respon Alat terhadap Sensor Cahaya</i>	Skenario 1	Berhasil
			Skenario 2	Berhasil
3	UC-003	Pengujian Keberhasilan Aplikasi <i>Mobile</i> dalam membaca data pada sensor	Skenario 1	Berhasil
4	UC-004	Pengujian Keberhasilan Aplikasi <i>Mobile</i> dalam membaca data pada sensor	Skenario 1	Berhasil
			Skenario 2	Berhasil

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diperoleh selama pengerjaan Tugas Akhir dan saran mengenai pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

5.1 Kesimpulan

Dari hasil pengamatan yang dilakukan selama proses perancangan, implementasi dan pengujian yang dilakukan dapat diambil kesimpulan sebagai berikut :

1. Sistem jemuran pakaian otomatis berhasil dibuat menggunakan *microcontoller* Arduino beserta komponen-komponen yang tersusun di dalamnya yaitu berupa *Motor Gear DC*, Saklar *Microswitch*, Sensor Hujan (*Raindrop Sensor*), Sensor LDR (*Light Dependent Resistor*) dan Sensor Kelembapan (*DHT11*). Hal ini ditunjukkan dengan berhasilnya pengujian respon sistem terhadap rangsangan yang diberikan pada sensor.
2. Aplikasi *mobile android* untuk memantau kondisi cuaca di rumah berhasil dibuat dengan menggunakan *MIT App Inventor* ditunjang dengan *modul wifi ESP8266* serta menggunakan *Thinkspeak* sebagai *server*. Penentuan kondisi cuaca dilakukan menggunakan *fuzzy logic*.
3. Aplikasi *mobile* untuk mengontrol jemuran secara manual berhasil dibuat dengan menggunakan *MIT App Inventor* ditunjang dengan *modul wifi ESP8266* serta menggunakan *Thinkspeak* sebagai *server*.

5.2 Saran

Berikut merupakan beberapa saran untuk pengembangan sistem pada masa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang dilakukan :

1. Untuk menerapkan Sistem Jemuran Pakaian Otomatis dalam kehidupan diperlukan adanya kekuatan torsi motor yang lebih kuat dan *power supply* yang lebih kuat pula voltasenya, semisal *accu*.
2. Disarankan untuk penelitian selanjutnya menggunakan *modul wifi* yang sudah tertanam pada *board microcontroller* untuk menghasilkan keluaran yang stabil dan memudahkan pengguna untuk melakukan *coding* pada sistem. Hal ini disebabkan karena untuk menyambungkan *modul wifi ESP8266* menuju *Access Point* hanya bisa dilakukan dengan menggunakan *AT Command*.
3. Sistem Jemuran Pakaian Otomatis ini nantinya bisa dikembangkan lagi. Tidak hanya untuk menggerakkan jemuran pakaian akan tetapi bisa untuk mengangkut ikan, hasil pertanian dan lain sebagainya.

DAFTAR PUSTAKA

- [1] B. M. Dr. Edwin Aldrian, *Iklim Semakin Tidak Menentu, Dari Pemanasan Global Menuju Perubahan Iklim*, 2011.
- [2] "reichelt," [Online]. Available: <https://www.reichelt.com/de/en/Single-Board-Microcontroller/ARDUINO-UNO-DIP/3/index.html?ACTION=3&GROUPID=8243&ARTICLE=154902>. [Accessed 2 May 2018].
- [3] "elektrojo," [Online]. Available: <http://www.elektrojo.com/home/1520-dht11-temperature-relative-humidity-sensor-module.html>. [Accessed 2 May 2018].
- [4] [Online]. Available: https://wiki.eprolabs.com/index.php?title=Raindrop_Sensor. [Accessed 2 May 2018].
- [5] M. A. Prasetyo, "BoArduino," [Online]. Available: <http://www.boarduino.web.id/2015/08/konfigurasi-esp8266-sebagai-client-dan.html>. [Accessed 13 March 2018].
- [6] V. Michalis, "ArduinoAutoMotive," [Online]. Available: <http://www.ardumotive.com/how-to-use-the-raindrops-sensor-moduleen.html>. [Accessed 26 February 2018].
- [7] R. Santos, "Random Nerd Tutorials," [Online]. Available: <http://randomnerdtutorials.com/esp8266-controlled-with-android-app-mit-app-inventor/>. [Accessed 2 May 2018].
- [8] A. Pamungkas, "Pemrograman Matlab," [Online]. Available: <https://pemrogramanmatlab.com/2015/08/24/logika-fuzzy-untuk-sistem-pengatur-kecepatan-mesin/#more-1012>. [Accessed 30 June 2018].

- [9] K. Wardana, "Tutor Keren," [Online]. Available: <https://tutorkeren.com/artikel/tutorial-lengkap-menggunakan-fuzzy-logic-toolbox-pada-matlab.htm>. [Accessed 30 June 2018].
- [10] "Blog Zerokol," [Online]. Available: <https://blog.zerokol.com/2012/09/arduinoofuzzy-fuzzy-library-for-arduino.html>. [Accessed 1 July 2018].
- [11] "MIT App Inventor," Massachusetts Institute of Technology, [Online]. Available: <http://appinventor.mit.edu/explore/resources.html>. [Accessed 21 May 2018].
- [12] "Thinkspeak," [Online]. Available: <https://thingspeak.com/>. [Accessed 2 May 2018].
- [13] "LantaiKayuIndonesia," [Online]. Available: <https://www.lantaikayu.biz/harga-lantai-kayu/harga-lantai-kayu-jati-2016.html>. [Accessed 2 May 2018].
- [14] "IyungTux," [Online]. Available: <https://iyungtux.web.id/2018/03/list-harga-harga-kertas-stiker-graftac-terkini.html>. [Accessed 2 May 2018].
- [15] "Open Bulids Part Store," [Online]. Available: <https://iyungtux.web.id/2018/03/list-harga-harga-kertas-stiker-graftac-terkini.html>. [Accessed 2 May 2018].
- [16] "Just Air Compressor," [Online]. Available: <https://justaircompressor.com/replace-the-timing-belt/>. [Accessed 2 May 2018].
- [17] Lumitha Seema Cutinha, Manasa K, Venkatesh Pai, Sadhana B., "Automatic Cloth Retriever System," *Automatic Cloth Retriever System*, 2016.
- [18] "Instructables," [Online]. Available: <http://www.instructables.com/technology/arduino/>. [Accessed 25 February 2018].

- [19] A. Sanjeev, "Maker Pro," [Online]. Available: <https://maker.pro/arduino/tutorial/how-to-use-an-ldr-sensor-with-arduino>. [Accessed 26 February 2018].
- [20] M. Manohar, "Maker Pro," [Online]. Available: <https://maker.pro/arduino/projects/arduino-weather-station>. [Accessed 26 February 2018].
- [21] D. Nedelkovski, "How to Mechatronics," [Online]. Available: <https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/>. [Accessed 13 March 2018].
- [22] M. A. Prasetyo, "BoArduino," [Online]. Available: <http://www.boarduino.web.id/2016/01/update-firmware-esp8266-dengan-arduino.html>. [Accessed 13 March 2018].

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Penulis, **Adetiya Bagus Nusantara**, lahir di Kediri, 2 November 1996. Penulis menempuh pendidikan sekolah menengah pertama di SMP Negeri 4 Kediri dan melanjutkan pendidikannya di SMA Negeri 2 Kediri. Selanjutnya penulis melanjutkan pendidikan sarjana di Departemen Informatika, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember Surabaya. Selama kuliah, penulis aktif menjadi santri pondok di Pondok Pesantren Khoirul Huda I Surabaya dan aktif pada beberapa kepanitiaan seperti LKMM Pra-TD, Schematics, dan kegiatan lainnya. Dalam menyelesaikan pendidikan S1, penulis mengambil bidang minat Arsitektur dan Jaringan Komputer (AJK). Penulis dapat dihubungi melalui email: bagus.nusantara313@gmail.com.